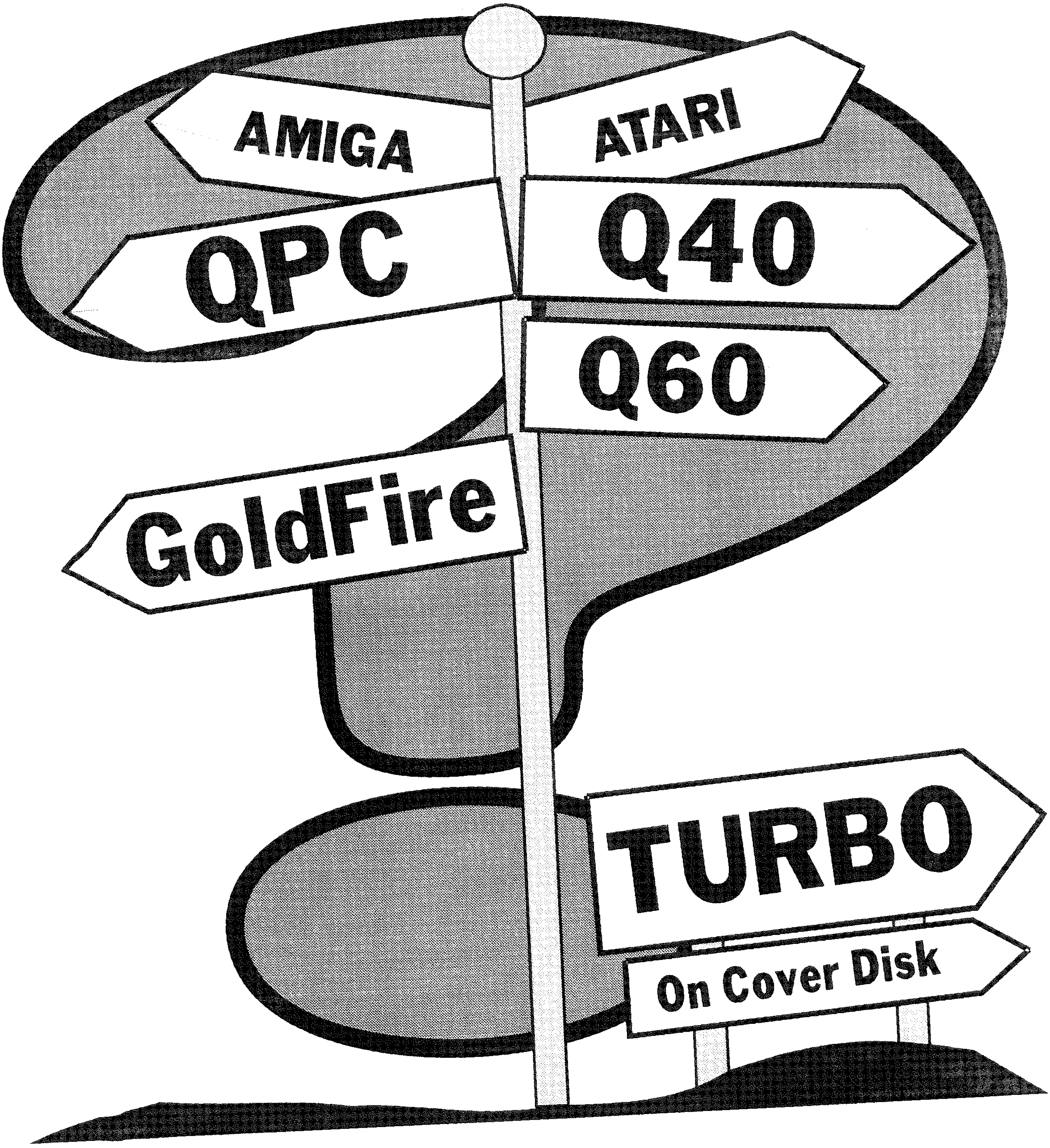
QL Today

Volume 5
Issue 5
Jan./February
2001

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...



Contents

Advertisers

in alphabetical order

Jochen Merz Software	35
QBOX USA	23
QBranch	14, 15
Quanta	29
RWAP	51
W.N. Richardson (EEC)	47
TF Services	43
Geoff Wicks	39

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 1708 510764
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is ©copyright 2000 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Editorial

Bruce Nicholls

Welcome to another year of QL activity.

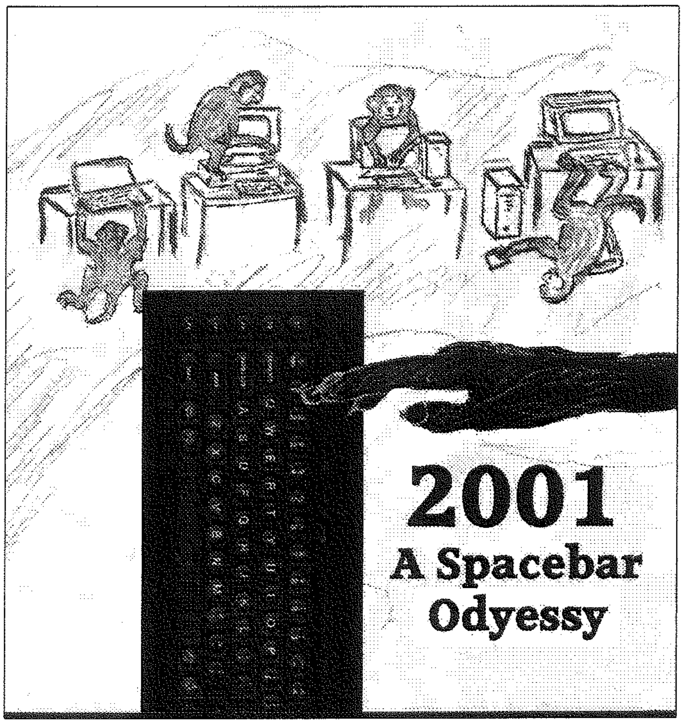
Last year saw the culmination of several high profile projects for the QL, including the much anticipated colour drivers and TCP/IP. We now have to build on these this year to provide more system software that can utilise these new features but is still compatible with 'older' systems. I know a lot of work is being done in the background to allow this to take place and so we can look forward to a productive new year.

As detailed in this issue you can now write Pointer programs with the now 'freeware' Turbo that has been updated by George Gwilt and is now compatible with SMSQ. The fabled Q60 is not a myth and is producing striking benchmark results, a must for those who want the ultimate QL power machine thanks to Claus & Peter Graf.

Marcel Kilgus has produced the final release version of QPC2 Version 2 which now comes with an extra bonus of native file support, this allows the reading of the host computers harddrive using a standard QL device so typing in "dir DOS1_ " shows all the files on the host machine's C: drive, as this is presented as a 'normal' QL device most software will be able to utilize this and it will be used a lot in the QLToday production office. Jonathan Dent is refining TCP/IP for 'normal' QL's and compatibles at present, as detailed in his article, and you can now exchange Internet emails via the QL.

We continue several of our 'regular' columns including QLTdis and programming Prowess [As we go to press a new version of Prowess has just been released to enhance it's font support]. I hope these articles inspire you to go and delve deeper into guts of the QL and if you have any experiences in using the information within these articles we would like to know.

Cartoon



NEWS

Dilwyn Jones Website

TurboPTR and T_Config

I have just added two Turbo compiler utilities from George Gwilt to my website:

1. TurboPTR (approx 300KB download) allows you to write pointer driven programs which can be compiled using Turbo. At last! Since QPTR toolkit extensions needed to be able to pass parameters to machine code procedures by reference and also to pass array parameters, George has recreated similar extensions as BASIC procs/fns to allow their compilation by Turbo.
2. T_Config (approc 30KB download) lets you write config blocks which can be included in Turbo compiled programs.

In addition, the version 4.4 update of Turbo compiler and version 3h27 of the Turbo Toolkit can be downloaded from the same site, along with Tim Swenson's updated Turbo manual. The new turbman_zip Turbo manual is a 246K download.

To download these, go to my Other Software Page **http://www.soft.net.uk/dj/software/other/other.html** and click on the Turbo link at the top of the page, TurboPTR and T_Config are just below the Turbo compiler download link on the (admittedly long) page.

This issue's cover disk will bring the latest Turbo to all QL Today readers with and without internet access.

LEAR PCBCAD 6

Malcolm Lear has kindly sent me an updated version of PCBCad 6 (a printed circuit board and CAD program, formerly commercial software) which has been placed on my website for you to download.

www.soft.net.uk/dj/software/other/other.html

Suqcess Demo version

Wolfgang Uhlig has released a trial version of his excellent Suqcess database manager program. This trial version is available on Thierry Godefroy's Web site download page

http://qdos.cjb.net/english/download.html

- go into the "Demo version of commercial software" section.

QUANTA CD-ROM

Roy Brereton has let us know that the Quanta software library on CD Rom is available from him for a cost of £10 including post and packaging. Purchasers would need to be able to read media with QXLWIN systems (i.e. QXL, Q40, QPC, uQLx or QemuLator 2 users), although there are programs such as Jonathan Hudson's WXQT2 allowing other computers to access QXLWIN formatted media. This CD-ROM is only available to Quanta members at the moment, though membership itself only costs £14 for British QLers and £17 for overseas QLers, so it may be worth that amount just to get the monthly newsletter and access to the entire software library on one CD for just £10!

OOPS! HTML Tutorial

We slipped up and gave the wrong address for Norman Dunbar's HTML Tutorial series of articles in the 'Are We Ready For The Net' article in the last issue. The correct address is:

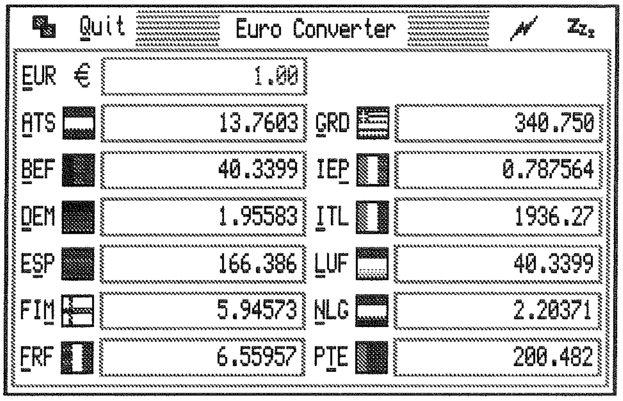
http://www.dunbar.cwc.net/html/index.html

Since then, Norman has been a busy little bee and added tutorials on simple forms and even a little Javascript!

EURO Converter Update

Andrea Carpi tells us that on the Beginners' Club (Italy) web site you can now find the new version 1.30 of their Euro converter program, now with GRD (Greek currency)

http://www.geocities.com/beginners.geo/ e04e.htm



Euro Converter

EUR €	1.00		
ATS	13.7603	GRD	340.750
BEF	40.3399	IEP	0.787564
DEM	1.95583	ITL	1936.27
ESP	166.386	LUF	40.3399
FIM	5.94573	NLG	2.20371
FRF	6.55957	PTE	200.482

JMS NEWS

Marcel Kilgus, author of QPC, was busy as promised and - QPC2 Version 2 Final release is ready!

He has also added a bonus for the delay. You can now access DOS and Windows partitions of your harddisk DIRECTLY, just type DIR DOS1_ and you get the directory of C:, DOS2_ is D: etc. The DOS device driver also appears as a device called

DOS in applications such as QPAC2, so that you can read files directly from a DOS formatted hard disk directly from the QPAC2 files menu for example.

You can download the update (if you already bought or upgraded to QPC2 Version 2 - you need a password) from the updated website
http://qpc.j-m-s.com

It is also easier now to reach the Jochen Merz SMSQ/E homepage by typing
http://smsq.j-m-s.com

News From Arnould Nazarian

For those interested by the differences and advantages of QDOS-SMSQ/E-Stella architecture there is some news.

In south of France, Montpellier, there is a company named Nexwave, **www.nexwave-solutions.com** I had found them by chance during summer 99 and I visited them with Tony Tebby (we were in Montpellier to visit Palm Computing).

It turned out to be a very interesting visit. A team of 3, 2 salesmen, 1 programmer, try to reinvent system software. And what they do is very close to what TT described in his document about Stella. For those interested, there is a white paper that explains what they are doing.

Interestingly there is a very tiny link between the QL and their activity. They come from the Amiga world. Emmanuel Marty, their programmer, seems to have learned a lot on this computer. And you maybe remember that the first OS of the Amiga was done by Metacomco because Metacomco had a good experience of the 68000 processor due to QL... (OK I wrote a _tiny_ link)

Emmanuel is in the same league as TT. He has compiled a web page about OSes and he has put some comments about Stella into it. It is at
http://tunes.org/Review/OSes.html

Ah yes, my idea was that Tony Tebby could work with them. But this has not happened until now. And I don't know why.

EASYBASE 0.56

Version 0.56 of EasyBase database system is now available from Q-Celt Computing. The update cures a problem in setting screen mode when running on a Q40 SMSQ/E system.

9th Italian QL Show

Davide Santachiara writes: You may find the link for full photo report of the 9th QL Italian meeting on my web page:
http://get.to/ergon

QL Users Mailing List

Thierry Godefroy writes: As of 31 december 2000, I will cease to archive the QL-users messages on my web site.

I registered both QL-users and QL-developers mailing lists into "The Mail Archive" web service
http://www.mail-archive.com/

QL-users archives are already available from:
http://www.mail-archive.com/ql-users %40nvg.ntnu.no/
and QL-developers will be soon (i.e. as soon as new messages will be sent to the list) at:
http://www.mail-archive.com/ql-developers%40nvg.ntnu.no/

News from J. Grimbert

Jérôme Grimbert's web site is now at
http://grimbert.cjb.net/
(this is a redirecting URL to the possibly moving site). Direct access to the QL part can be done with
http://ql.grimbert.cjb.net/

News from Dave Westbury

After a long break due to work I have picked up the Photon JPEG viewer project again. The new version uses a more efficient IDCT algorithm. It has not only reduced the file size by 50% (now only 7750 bytes which includes the viewer and Floyd Steinberg Error diffusion dithering!) but is 2 to 3 times faster on a Q40 (and hopefully QPC/QXL). The Aurora performance is also quicker but is still slugged by heavy dithering code which, along with some speed improvements in colour space conversion etc, I shall be looking at next. I have already sussed out the GIF stuff (incl. correct animation framing) so this should follow soon. Photon is now easily the fastest QL JPEG and works on any machine in any mode :-)

MySQL on Q40-LINUX

Claus Graf writes: I've just compiled MySQL 3.22.32 for Q40 Linux. This is a basically free SQL database server (it's ***not*** free for M$ windoze machines, another reason not to use Gate$ products).

I have chosen the source RPM from a Suse 6.4 for PPC and it compiles without any problem. Simple commands like creating databases, create table, select, delete and update are tested and work.

QL in Personal Computer World!

One of the best known British PC mags, Personal Computer World, has chosen to carry an article about the QL! One of their editorial team got in touch with Roy Wood and then wrote an article about the QL in their Leisure Lines column in the February 2001 issue which will have been out for some time by the time you read this. They included the address of the QBranch web site so hopefully some PCW readers (and ex QLers perhaps) will find out that the QL is still alive and popular as a result! I've tried sending them some material for publication to follow this up and it has been suggested that we try to get them to put a QL emulator on one of their cover CDs for maximum QL publicity.

QL Today News

We have kept the price for QL Today stable for many years. The quality of the magazine has improved, we deliver goodies like cover disks and calendars etc. but the rest of the world puts higher prices onto us. UPS has raised the shipping costs three times, printing costs have gone up by over 20% and the postage has gone up three times in considerable steps, especially to our readers in the USA. To maintain the size and quality of QL Today, we have to adjust the subscription rate slightly from next issue.

Postcards - the Result

The result is VERY disappointing. Even the majority of the QL Today readers decided not to return the postcard, let alone all the other QLers from our merged database. We would like to thank everybody who returned the postcard to us, but in the end we do not know what to do. Do we erase everybody from the database who has not returned the postcard? We would be left with 176 customers only! Shall WE select whom to delete and whom not? Impossible, we can't tell! In the end, our aim to reduce costs by mailing only those people who are interested has not been reached. We shall discuss what to do.

Anyway, we think it is worth mentioning how we drew the lucky winner:

```
RANDOMISE date: PRINT RND(1 to 176)
```

The result - 85. Tony Firshman then selected the 85th entry in his database file:

James Patterson
2 John's Road
EYEMOUTH
Berwickshire
TD14 5DX
United Kingdom

Congratulations! Just refer to this issue of QL Today, James, when you decide to redeem the voucher in an order with JMS or QBRANCH!

Hints N Tips

Q. After setting the QPAc2 Files menu to Sort By Name, if I wish to cancel the sorting, I bring up the Sort By menu (F6 or SHIFT F1) and press ESC. The Files menu then shows Sort By nothing, but the list of files displayed is still sorted.

A. Hit the refresh icon (the lightning flash symbol) or press CTRL F2 to refresh the list shown. Files reads the list of files again and this time it will be unsorted. This can also happen if you MOVE a file to a new name (i.e. rename a file on the same medium), that the new name may not be shown until the list is refreshed.

This one comes from David Denham:

Q. How can I delete a directory name in QPAC2 files menu. I have tried highlighting it, DO on it just moves up to that directory level rather than offering me the usual options to Copy/Move/Delete etc.

A. A do on a directory name will, as you say, cause the Files menu to move up to that directory level rather than offering the files context menu. To get around this, highlight (HIT) the directory name to be deleted plus one other name (a file or perhaps another directory is safer). Once two or more files are highlighted, DO (or press ENTER), select the Delete command and selectively delete the directory name required.

The next one is a query from J.C.Marcus:

Q. There seems to be a small "feature" of using ramdisks that does annoy me. That is, as far as I can tell, there is no way of recovering memory used by a ramdisk, even after deleting the files contained in it. This means that if I do a lot of backing up through RAM, that memory is lost for me, and can be a bit frustrating. I wonder therefore if there is any solution to this, or am I just missing something simple like RTM! I am using SMSQ/E 2.91 by the way, but from what I can remember, using ramdisks under PE always had this problem.

A. The answer is pretty simple: deleting the files releases the memory occupied by the files to the free memory, but not the "directory" of the RAM Disk. The dirctory is created as soon as you access a RAM Disk. When you FORMAT it, even the directory is removed but as soon as you ***look at it*** it will be created again. So, don't look at it after FORMAT, trust it to be gone!
Compare this with a floppy disk: when you FORMAT it, you get 1440 of 1440 or 2880 of 2880 sectors. As soon as you DIR it, you will see that some of the sectors are gone - used for the directoy (even though it is empty) and the floppy disk map. There is no way to get 1440 sectors back until you FORMAT the disk again, but s soon as you look at it, you will loose the sectors. The diffrence between RAM Disk and Floppy disk is, that the map for the floppy disk will be created immediately after FORMAT, the RAM disk will only use up space when it is accessed first time.

The Cover Disk

As promised, Dilwyn has packed most of the Turbo goodies onto one disk. To make sure that all QL Today readers can read the disk, we had to put it onto DDs ... one part is missing, but the next QL Today cover disk will deliver this to you next issue.

The disk contains three packages which can be unpacked very easily. Just LRUN the BOOT program or insert the cover disk and press the RESET button.
A menu will appear, giving you the choice of the three packages:
1. The Turbo Manuals
2. Turbo Compiler and Toolkit
3. T-Config (Config Utility)

Select the package you want to unpack by entering 1, 2 or 3. Then enter the drive containing the floppy disk you want to unpack from (this should be flp1_).

Finally, enter the device you want to store the files from the package in (for example, another floppy disk flp2_ or a RAM disk ram1_ ... ram8_).
The rest is automatic - you will find the following files on the destination device:

For package 1:

```
README
TURBODEM_TXT
TURBOREF_TXT
TURBOREF1_TXT
TURBOREF2_TXT
TURBOREP_TXT
TURBOS1_TXT
TURBOS2_TXT
TURBOS3_TXT
TURBOS4_TXT
TURBOTK_TXT
TURBOTOC_TXT
INTFILE_TXT
LINKLOAD_TXT
TEST_BAS
UPDATE_TXT
```

For package 2:

```
TURBO_CONFIG
TURBO_SMS_CODE
TURBO_TK_CODE
UPDATE_TXT
CODEGEN_task
PARSER_task
CONFIGURE
DEMOS_BAS
MiniCommdLin_BIN
NewTurboTK_TXT
Present_EXN
ReSize_BAS
SMSQCommdLin_BIN
TOOLKIT_CONFIG_BAS
TURBO_CONFIG_BAS
TurboFix_BIN
TurboPatch
TurboPatch_TXT
UTILITY_TASK
```

For package 3:

```
readme
t_config_data_bas
T_CONFIG_LOAD_BAS
T_CONFIG_DATA
t_config_text
T_CONFIG_LOAD
CONFIG_INS
```

Please read the text files containcd in the packages first!

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Jonathan Hudsons web site where you find more information about lots of interesting QDOS software and INFOZIP at
www.bigfoot.com/~jrhudson/

Gee Graphics! (on the QL?) - Part 20

H L Schaaf

Using the Inverse of a matrix

Last time we used Cramer's rule to solve simultaneous linear equations, this time we'll merge in a short and sweet method that makes use of the Inverse of a matrix.

I had the pleasure of meeting John Sadler at QL2000. He took a look at the matrix routines and found ways to simplify them. The listing Sadler_bas is a snippet from a longer listing that he kindly sent me via e-mail. I thank him for his contributions and hope he will continue to make them.

To see the Inverse approach in action:

Load the matrix routines "MatFunPROCs_BAS" with the changes mentioned in GG#19.
Then merge the listing "Cramer_bas".
Then merge the listing "Sadler_bas" (listed below)

```
900 REMark Sadler_bas
910 PAUSE 100 : REMark to go with GG#20, HL Schaaf, December 15, 2000
920 PRINT \\"The advantage of matrices is we can use matrix algebra to solve a"
930 PRINT "series of equations because if [U]x[X] = [C] then [X] = 1/[U]x[C]"
940 PRINT "So solution = the inverse of the coefficients times the constants."
950 PRINT \"Hence constants ";
960 PRINT ,C(1 TO n),\
970 PRINT "Produce solution ";
980 MAT_INV U, W
990 MAT_MUL W, C, A
1000 PRINT ,A(1 TO n),\
1010 PRINT \"Also the constants ";
1020 DATA 12, 5, 18
1030 FOR i = 1 TO n
1040   READ C(i)
1050 END FOR i
1060 PRINT ,C(1 TO n),\
1070 PRINT "Produce solution ";
1080 MAT_MUL W, C, A
1090 PRINT ,A(1 TO n),\
1100 PRINT \"And the constants ";
1110 DATA 24, 3, 14
1120 FOR i = 1 TO n
1130   READ C(i)
1140 END FOR i
1150 PRINT ,C(1 TO n),\
1160 PRINT "Produce solution ";
1170 MAT_MUL W, C, A
1180 PRINT ,A(1 TO n),\
1190 PRINT \"Which also runs faster for the other solutions as well!"
```

The Inverse method seems to get the about the same answers but in less time. In addition it is easy to change constants and then find a new solution for the same coefficients.

To see what the Inverse matrix looks like just type "MAT_SHOW W".

Next time I hope to explore a "Gaussian elimination" approach.

Q60 Benchmarks

from Peter Graf

As requested by some speed-hungry QLers, I tried some benchmark testing with my Q60 prototype. Indeed the results are interesting, so I have decided to publish them, although the Q60 has not gone into production yet. Unfortunately, the question when the Q60 can be bought, can't be answered yet. There are still problems with SMSQ/E, for example the use of the 68060 caches and the missing MOVEP command. QDOS Classic already runs quite good on the Q60 and Linux seems to run almost without problems.

The tested prototype is a Q60/80 with a 68LC060RC75 CPU that runs at 80 MHz. Except the faster CPU with bigger caches and the modified memory interface, the concepts are similar to Q40. That means a QL-compatible mainboard with highcolor-graphics, stereo-sound, extension bus for ISA cards, connectors for the usual serial/parallel interfaces, harddisk, floppy and PC-keyboard.

I executed the four benchmarks which are probably most frequently used in the QL sector.

Bogomips: An indicator how many instructions can be processed per second. Worldwide one of the wellknown benchmarks. Version 1.5 from Thierry Godefroy was used. For results see diagram 1.

①

GoldCard	1,61
SuperGoldCard	5,78
Athlon 600 MHz (Windows NT) QPC 2	11,01
Q40	26,63
Q60/80	160,02

Diagram 1: Bogompis

Dhrystones: A synthetic benchmark, that is meant to be represantive for the integer-system performance. The results depend on the programming language. The version 2.1 from Thierry Godefroy compiled with the C-compiler GCC was used. For results see diagram 2.

②

SuperGoldCard	6667
Athlon 600 MHz (Windows NT) QPC 2	17482
Q40	36443
Q60/80	101010

Diagram 2: Dhrystones/s

QSBB: A BASIC-benchmark, that was presented by Al Feng in QL Today July/August 1998. The results for QL and QXL have not been measured by myself, but copied from this article. Attention: All values have been measured under SMSQ/E, except for the original Sinclair QL. This is unfair towards the QL, because BASIC is of course a lot faster under SMSQ/E than under QDOS. For results see diagram 3.

③

	Print	Function	String
Sinclair QL (QDOS)	980	840	1100
GoldCard	3920	8480	20750
SuperGoldCard	7840	23700	59440
QXL	14300	27900	70640
Athlon 600 MHz (Windows NT) QPC 2	7460	65060	146580
Q40	53000	99320	249640
Q60/80	103040	317300	852400

Diagram 3: QSBB

TEST909: A speed test for the operating systems QDOS, SMSQ and SMSQ/E by Rolf Ritter. It includes loop-test, maths-test, text-test, scroll-test and graphics-test. The table shows the overall results as a factor compared to the original QL. Attention: The result for the Q60 was measured with copyback cache switched off, without scroll-test, because there were still operating system problems here. With copyback cache switched on, a better result can probably be expected. For results see diagram 4.

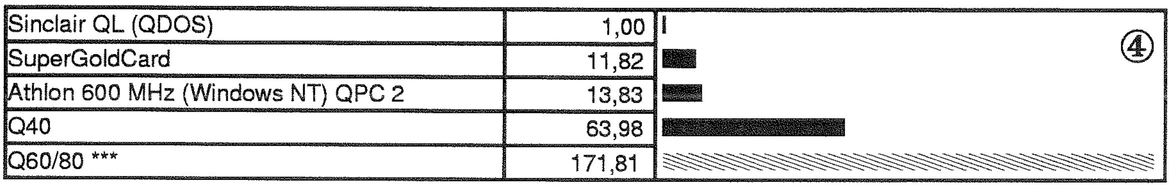


Diagramm 4: TEST909

The following QL-compatible hardware was used for reference:

Original QL, 128 KB, JS ROM
GoldCard, SMSQ/E 2.89
SuperGoldCard, SMSQ/E 2.89
Q40, 16 MB, SMSQ/E 2.92
Q60, 80 MHz, 16 MB, SMSQ/E 2.92 / QDOS Classic beta p

Additionally, the following systems with emulator have been used: PC with QXL, 20 MHz, result from QL Today July/August 1998, details not known
PC with 600 MHz Athlon processor, 196 MB RAM, Microsoft Windows NT 4.0, QPC II Demo Version 1.5, 512x256 full screen mode

Under Windows 95 I have not been able to test QPC II, because the PC crashed directly after the welcome screen. Maybe a problem with the Windows graphic drivers, which I had recently updated. To reproduce the benchmarks on Q40 and Q60 it should be noticed that I used SMSQ/E 2.92. More recent versions seem to have the copyback caches disabled by mistake, I hope this will be fixed soon. Under QDOS Classic it is possible to enable the copyback cache with the BASIC command `COPYBACK`

I know that benchmarks are not always representative, because every user has his/her own way of using a computer. Nevertheless the measured speed of the Q60 has surprised me. The Q60 is by far the fastest QL compatible computer of all time and also one of the fastest existing 680x0 systems. Even though my test was only based on my own prototype, I hope I have not bored anybody with it.

Q40 or QPC

Wolfgang Uhlig

In issue 3 of QL Today an article on this subject by Peter Jäger caused considerable commotion. The publisher was so angry that he gave a spontaneous and somewhat irritated reply in the same issue, and in the following German edition of QL Today Peter Graf gave an heartfelt and emotional defence. ***[Ooops, correction - the publisher was NOT angry - Publisher]***

As will be clear by the end of this article, my reply to the question is definite. I want to write about things, which were more than clear in the articles, and which have irritated me for some time.

In spite of the danger that of making myself completely unloved in the QL community, I have to say that the permanent, monotone and completely polarised idea of here the Holy QL Grail and there the wicked PC increasingly irritates me. It is true the Microsoft operating system has "conquered" the world and it is certainly not an example of reliability. But in this "evil" Windows world there are numerous inspired programmers who do their very best to ensure you see as few MS-bugs as possible. Marcel Kilgus, who has given us QPC, is a perfect example.

Ironically the Microsoft operating system has undoubtedly ensured that the computer has spread throughout the world in a lightening tempo, a development from which many people have profited or at least have had a lot of pleasure. Most people just want to play games, look at and edit their photos, gather a music collection together and listen to it, exchange information, chat or email in the simplest possible manner. They are offered this in plenty, free- and shareware programs are available like sand, every computer magazine comes with a CD full of programs, no wish remains unfulfilled and that at scarcely any cost.

Occasionally the computer crashes! So what?

The people who are upset by this are US, the purists, the hackers, the QL-ers! The trouble is the world is not upset by us, they could not care less how we tinker with our "stone age instrument".

It is ridiculous to scream that the chips just become faster, the hard disks and operating systems just get bigger, and that there are always new peripherals. You could equally complain that people now prefer to ride a car than a stagecoach. (If Motorola had not developed the 68040, where, Peter Graf, would be the Q40? Would you then say: "...yes, only faster, bigger and so on!"?) The world is like that and will not change. So, if you can't avoid it, why not enjoy it?

Contrary to Peter Graf I don't think that the PC owner must buy a new machine at every least opportunity to remain up to date. (In 8 years I have bought only two new PCs, the last one was a 400Mhz processor a year ago that I upgraded for a song. This is more than enough for the use I make of it and QPC works fine on it.) A lot of people must have a PC for their work; others like to work with photos, layouts and large, complex texts and they cannot manage without. For these people the present normal processor of 750 Mhz is fast - and even for the future - sufficient. Clock speeds of 1Ghz are becoming the norm. 3 - 5 Ghz processors are in the pipeline for 2002. A fantastic outlook for anyone who can be a little patient and a fantastic outlook for QPC users!

I hear repeatedly from the QL side that the MS operating systems are bad, but ours are elegant, userfriendly and reliable. A QL-emulator is thus no solution, but it is better to use the original (now the Q40). Apart from the fact that I have had a different experience during all my QL years, I find it arrogant to think we can cold shoulder the world and retreat into our "stand alone" QL system. It is also unbelievably stupid.

If we will really look in a direction that gives a chance of survival to the QL, then please don't let us do it like a pouting child hiding in his bedroom! If someone not familiar with the QL gets a chance to know it, then it is not because he has bought a Q40. Equally no one will buy a Q40 purely because it can also run Linux. That's just wishful thinking.

The only chance the PC owner will get to know the QL is a QPC, a good description of how to use it and software that makes it attractive. Wouldn't it be lovely with a chic modern program that would run safely in Windows to persuade someone "out there" that there are alternatives!

But this, in my opinion, is our biggest problem; in the last few years there has been too little effort in the development of new and modern software. In comparison with all the efforts and attention to new hardware, the program booty is scanty. I exclude TT's fantastic colour drivers because they are not a program.

Some good programmers have departed and there are few replacements. Important tools such as EasyPtr are no longer being developed, a disaster for Basic programmers. At the moment I cannot see any change in this and that leaves me sceptical about the future of the QL. Everyone who knows me, knows I am one the most devoted QL enthusiasts. That was so in the past and still shall be. It is not for nothing (or perhaps for nothing :-)) that over many years I have written a summary of the English QL Today for the German QL Today as well as EasyPtr tutorials and other articles. I also attempt to interest other people in the QL. I have written a complete program for a bicycle shop in the town where I live. It has three network places (QXL-netten) to enable cashdesk, workshop and office programs to work together. Many customers have asked what "Windows Version" it is :-). And finally I have programmed SUQCESS just to oppose that!

What I have written is absolutely not a defence of windows, but an attempt to correct the perspective.

I fully respect the achievements of Peter Graf in developing and creating the Q40, but I think only the QPC coupled with new and modern software will create the chance for the QL to survive and perhaps, just perhaps, even to grow. That is what I want!

In Bed with the Mistress

Geoff Wicks

We QL-ers are highly promiscuous. We like to think of ourselves as pure, unsullied and modest, but, although we have promised to remain eternally faithful to our partner, we regularly go to bed with the mistress. Indeed, if the truth be known, were not for the mistress, the partner would have disappeared long ago.

The evidence is all around us. Take a look at the internet. The QL is well served with websites, some of which are of high quality. Much of the most recent freeware and public domain software is almost impossible to obtain without access to the web. Maintaining a website, particularly a high standard one, is a time consuming process, which, in spite of the best efforts of Jonathan Dent and others, cannot yet be done from a QL. These websites mean QL users are spending hours at the keyboard of a PC.

Then take a look at the traders. Most of our contact with one another is done electronically, which is cheaper and quicker than snail mail. QL Today relies heavily on email and most of the copy, both articles and adverts, is received electronically. QL Today is based, editorially and in distribution, in both the UK and Germany, which would be almost impossible if everything was done by post. Soon 'online QL-ers' will receive notice of shows via email rather than by flier. Even the "tired looking old men" of QUANTA now have their own website. QL-ers are spending a lot of time using PCs.

QL Today readers, who subscribe to the ql-users email group, will recognise the unpleasant jibe of "tired looking old men" from comments about QUANTA. The ql-users group corresponds electronically about all QL matters, including hardware, software, gossip and jokes. Usually there are under 20 contributions a day, but just occasionally it goes into what I call "shark feeding frenzy mode" and the messages can leap up to 50 or more. Quite often there are derogatory comments about PCs in general and Microsoft in particular, but to my knowledge only one subscriber actually uses a QL to access the group. Most of the rest are using PCs. I know in some cases it is the boss's PC in the lunch break, but it is still using a PC to further QL interests.

Nevertheless, the antipathy of QL users to PC's takes concrete forms. The QXL card never had the success of the Gold and Super Gold cards, and in spite of the dire predictions of the pessimists, there is still an interest in new QL hardware. There are many who believe the Q40 arrested the steady decline of the QL, and gave an impetus to the development of new colour drivers and other software. There are many who profess a hate for the PC and who claim only to be happy when it is running QPC. I can almost hear them say, "I always wear rubber gloves and keep my eyes firmly on the ground until QPC is loaded".

The QL community has proved to be remarkably resilient, although it is a fragmented community with practically every QL user doing his own thing. All the same, I think there are two distinct streams, the hardware stream and the software stream. What both streams have in common, is a dislike for the herd instinct and a need to express individuality. To whichever stream you belong, you do not have to be an "expert" to fit in with the QL community. You can still do your own thing. The nature of the QL means that in whichever of the two streams you place yourself, you also have to dabble in the other stream.

QL users have to be realists. To survive in the big world outside the QL you have to use PC's. We need both our partner and our mistress. Perhaps we love the partner, but feel the pressure from the mistress.

Of the two QL streams I am firmly in the software one. For this reason perhaps I have chosen the PC rather than the native hardware way. My main QLs are a QXL card and a QPC laptop, although I still have a JM and Minerva black boxes for downward compatibility testing. How do I divide my attentions between the partner and the mistress? Most of my QL time is spent word processing. A seemingly innocuous statement that is, in fact, earth shattering. Word processing is one of the QL's great deficiencies. Our word processors are not a patch on those often now delivered "free of charge" with a new PC.

So why do I stick with the QL word processors? In one word "simplicity". It would be hard to find a more user friendly word processor than Quill, and Quill set the standard for Perfection and Text87. I use Text87 for most of my work, because it handles proportional fonts better than the other QL word processors. However the ease of loading and the simplicity and speed of the search and replace commands of Perfection are preferable when I write data bases for my commercial programs.

There is much I dislike about Text87, but I accept these disadvantages, because I can use it happily alongside the tools I have developed to assist me in creative thinking and writing. Many professional writers are highly critical of Microsoft Word, because its spell-checker, thesaurus and style-checker force then into a stereotyped way of thinking. The QL enables me to adapt its word processing facilities to my own needs.

I rarely need the extra features of the PC word processors. When I do it is because I need a complicated layout or design for, say, one of my manuals. Even then the bulk of the work is done on the QL before transferring to the PC for the final layout.

In the last year I have had much contact with former and current QL users, who needed help with transferring QL files to a PC. It is surprising the depth of work that has been done on the QL. One person is writing a book on Polish airmen in the 2nd World War; another has made a study of Arctic explorers; and a third is a retired professor who writes his theses using Text87 on a Trump Card black box. There are also many secretaries of clubs and community bodies who use their QL in their work. Some ex-QLers spoke nostalgically of their years on the QL, when they could do their own programming. How many of these people, or for that matter, how many QL Today readers would know how to start programming on the PC? Or even to adapt a PC program for their own use?

As yet I have not found any PC program to take the place of LineDesign. Many PC graphics programs are for people who wish to edit their own photographs and display them. The PC word processors allow you to design complicated layouts and posters, but there is nothing that I could use to prepare my adverts for QL Today. LineDesign enables me to quickly combine text and "artwork" and then convert them easily into a gif image using QUNPIC. Most PC programs only allow you to save complete pages in their own formatting, have limited file transfer facilities and no possibility to convert to a single gif or similar image.

And now the mistress. Why do I need a PC? In two words Communications and Contacts.

What I know about communications, I learnt mainly on a PC. My early experiences were gained using an internal modem in a PC when my ex-wife was participating in a teleworking experiment. I soon discovered the limitations of the PC world as it took several minutes to get a secure connection to her university, whereas I could access the QL BBS within a minute.

When an external modem became available, I had no desire to dismantle a QL and set it up again in another room or to go to the expense of a Super Hermes. Nowadays there is little point in running QL communications programs on a QXL or a QPC instead of native PC software. (However there has been a suggestion on the QL email group that it is easier to pick up a incoming fax on QFAX than on native PC software.)

Communications have become an essential part of my QL life, and play a vital role in keeping Just Words! alive. I am writing this article too near to the QL Today deadline for it to be sent by post. Increasingly I find people have downloaded demo versions of the programs from my web page before buying. Email also enables me to offer a better after sales service to customers. At the last Eindhoven show a customer reported a bug in QL-2-PC Transfer. By the following Wednesday he had a corrected version, by which time he had found another bug. By Friday, within a week of the show, he had yet another version correcting this bug. This would have taken weeks if done by post.

All my email knowledge and my email administration has been done on the PC, and in spite of the efforts of Jonathan Dent and others, I am unlikely to move to QL communications. Nevertheless I hope to play my part in actively encouraging email and internet use by those people who prefer to stick with native QL hardware.

Recently I have been conducting a lengthy email correspondence with a woman in the United Arab Emirates whom I "met" on the internet. Her 3x great grandfather and my 3x great uncle was one Richard Kirk, who in the 1830s and 1840s was the Town Crier of Chesterfield. He is one of my favourite ancestors, because in 1841 he was summonsed before the Watch Committee and fined 3 shillings for using his bell without permission to summon a meeting of Chartists, a group of political reformers, and had thereby "endangered the peace and quiet of the borough".

We have a lot of information to swap, which has to be done electronically, and she asked me which "GEDCOM" I used. I could probably have found a way of converting between ASCII text and Archive, but it is difficult to impose my way of thinking and working on a PC user. It is simpler to use the free copy of a genealogical program that came on a cover disk. I have known QL users who have had great difficulty in persuading their PC colleagues to save documents in ASCII format.

Five years ago, my main concern was always to have a backup QL for emergencies. Now I want both a backup QL and a backup PC. I do not see these as being in conflict with one another. Perhaps the time has come to stop our public protestation of faithfulness to the partner, whilst secretly cavorting with the mistress. I am in love with both.

EasyBase - A Review

By Tim Swenson

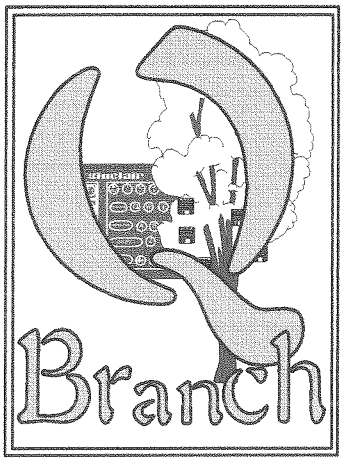
EasyBase is a new non-Pointer Environment database program written by Dilwyn Jones, and sold by Q-Celt Software.

The Manual

EasyBase comes with a substantial sized manual. The manual comes in three Quill _doc files. It covers all of EasyBase in good detail. Since the review copy of EasyBase was e-mailed to me, I don't know if there will be a hardcopy version of the manual.

The manual starts off with an introduction to databases and then jumps into configuring the program. The manual then goes through each of the EasyBase commands. As with most QL manuals, the writing style is light and informal.

Q Branch



PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

Big News this issue !

QPC 2 version 2 has now reached the final release stage and it is faster than any previous version. Marcel has not only increased the speed. He has added the 16-bit colour support and a device which allows you to copy to and from the DOS side of the PC s hard disk drive just as easily as if it were the QDOS partition. Put a real computer into your PC.

One other piece of good news is that Q Branch are now able to continue upgrading the Qubide ROMs to version 2.01. Phil Borman is making the code public domain and we have access to a programmer to blow the GALs and EPROMs.

A good start to the new year eh ?

We still have a few low priced superHermes available.

Full superHermes £ 65.00 with keyboard and mouse adaptors

superHermes Lite £ 30.00 with keyboard adaptor.

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00

(Includes Pointer and non-pointer driven versions)

(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)

Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

Spelling Crib : PD program £ 1.50 +SAE or Free if you buy all three programs

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v2.04	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

We are currently out of stock of the SuperBasic Reference Manual Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.

compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !

£ 40.00

+ postage

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

Please Note our new address and phone numbers

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor
Demo version £ 1.50 + postage
Full Registered version £ 18.00
Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	**£ 100.00**
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 50.00 *
Recycled Aurora	£ 75.00 *
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

14" and 15" monitors for the Aurora - Call.

SMSQ/E

Gold Card / Atari / QXL Version
£ 69.00
Various Atari versions : call for details

QPC 2 full colour version!

£ 82.00 (£ 67.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 35.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E or QPC 2 upgrade !

Q Branch Programs

The Knight Safe 3	£ 35.00
upgrades from previous versions	£ 5.00
Q - Route v1.08C Route finding programme	£ 25.00
Q - Count Pointer driven home accounting	£ 25.00

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

QPC 2 v2 FULL VERSION AVAILABLE NOW

Qubide upgrades to version 2.01 £ 8.00

VISA

We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.

MasterCard

Installation

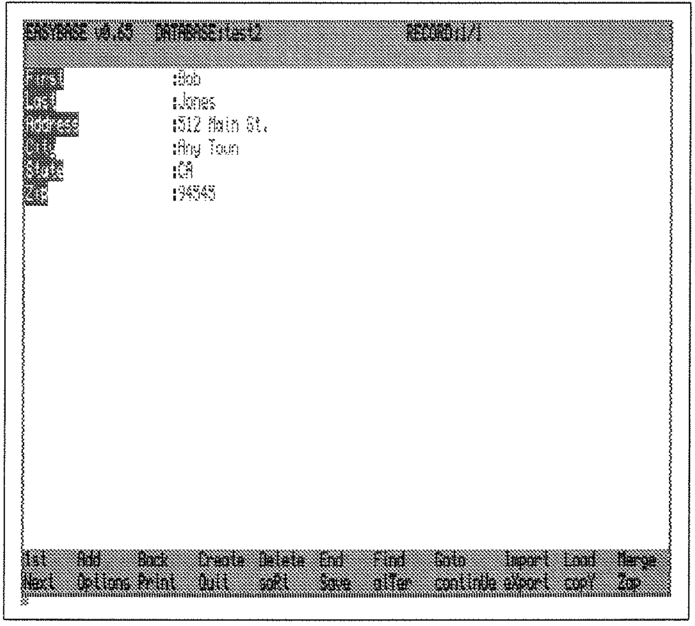
EasyBase does not require any Extensions to be loaded, so all of the files from the distribution diskette can be copied to a single subdirectory or floppy. The main executable comes with or without the Qlib runtimes linked in. A number of example database files also come with the application, allowing the user to jump right in with getting familair with the program.

Configuration

At first glance, configuration of EasyBase looks like a daunting task. There are 40 different items to configure.

All of the items are stored in a standard Config block and can be changed by using Config (a version comes with the distribution disk). A number of key items are configurable once the program is running.

The confguration items cover display, defaults, the printer and printing. Most of these items need be configured just once and then you are set for life (or until you change your hardware setup). The only one I changed was the default directory where the database files are located.



```
EASYBASE v0.65   DATABASE:test2                    RECORD:1/1
First                    :Bob
Last                     :Jones
Address                  :512 Main St.
City                     :Any Town
State                    :CA
Zip                      :94545

1st  Add     Back  Create Delete End  Find  Goto     Import Load Merge
Next Options Print Quit   soRt   Save alTer continUe eXport copY Zap
```

Once the program is running, a few items are configurable via the Options menu. The items are: Sound (on/off), Load directory, Save directory, default database file extension, default export file extension, and printer device.

Using EasyBase

EasyBase is started by just EXECing the main executable (either the _obj or _rtm version). The main EasyBase window then appears.

EasyBase works its display similar to Archive. The main screen is clear for output, the lower part of the screen lists the different commands and command prompts. Once a database is loaded, it is outputted to the display area, looking very similar to Archive.

Unlike Archive, EasyBase uses one-key commands. The commands listed on the bottom of the screen have one letter in upper case. This letter will envoke the command. For example, the Q key is the command letter for Quit. In most cases the first letter is the command letter, but not in all cases. Export (eXport) uses X for the command letter. Even though the command letters are in upper case, the command letters do not have to be (EasyBase is case insensitive on the command letters).

The types of commands are very similar to Archive (and most other databases). There are record based commands (first, next, sort, add, find) and file based commands (load, save, merge, import, export). The Create command is about the only table based command (although, one could view the file commands as table commands, as they act on tables). Of course there is also the Print command.

If you don't know much about databases, the manual gives a nice little tutorial and then leads the reader through each of the commands. Since most of the commands are pretty common to databases, someone familar with databases can just work through the commands without really studying the manual.

Some commands are immediate. In other words, they take affect without any more input from the user (1st, Next, Back, etc.). Commands that require more input from the user prompt the user for the information. All of the different options available to the user are displayed. Only answers that are reasonable to the question are allowed. All of this means that it's safe to explore the different commands, knowing that its hard to get somewhere where you don't know what to do next.

Creating a Table

Creating a table is relatively easy and the table creation screen lists all of the options and ways to navigate the creation screen. A table is created by giving all of the fields names. Then each field is given a field length (again, helpfull information is displayed at the bottom of the screen). Once this is done, the table name is entered and the table is created.

All of the fields are text and can store both text and numbers. This saves the trouble and complexity of defining the type of each field. Granted this has a limit to the speed of the database (easier to compare numbers over text) and the size of the database (numbers take up less space than text), but with the size of the databases used on the QL, neither of these limitations are a factor. EasyBase is not competing with Oracle, Informix, or Sybase.

Adding Records

Adding records to the database is done with the Alter command. The table is displayed with blank fields. You can then start entering new data in the fields. Hitting Enter at the end of each field takes you to the next field. When hitting Enter at the end of the last field, the new data is entered into the database and a new blank record is brought up. There are keys to move about the data entry screen, just in case you need to edit what you entered. Hitting Escape will take you back to the main menu.

Sorting the Database

Once you have a number of records entered, sorting the database might be a good idea. R is the command key for soRt. The first question for sorting is what type of sort; ascending or descending (called alphabetical and reverse). Hit 1 for ascending (alphabetical). Next you choose which field to sort on. Using the arrow keys you can scroll through each field, then hitting enter when you have selected which field to sort on. The database is then sorted and the first record is displayed when you are returned to the command screen.

EasyBase supports sorting on two fields. When selecting the field to sort on, hitting SPACE then takes you to another menu where you can choose the 2nd field ot sort on. This is a nice feature, but is not really needed.

Any database that supports sorting on any field also supports many subsorts. If you need to sort a database by last names, with a subsort on first names (os that James Smith will appear before John Smith), there is a little trick. First sort the database on first name, then sort it on last name. The first sort sorts those records with the same last name values. This same trick can be applied for any number of subsorts. Sort first the last field you want sorted, and then keep sorting on each more important field. This process can be time consuming, but it allows a database to be flexible and not fully support many subsorts in one sort command.

Finding Information

Adding and sorting information is nice, but the usefullness of a database is to find information. The Find command is done in three steps:

- enter the search string
- define the field(s) to search
- define type of match (exact, containing, starting with)

All of these steps are done in three menus. In defining the search field, you can select the field or take the default value of "All Fields", which searches all fields in the table.

Since there may be more than one record that matches the search criteria, the continUe command skips forward to the next record that match the search criteria.

Copying Information Into Other Programs

One neat feature of EasyBase is the Copy command. This command copies the contents of the current record (the one being displayed) into either the Stuffer Buffer or the Scrap. This allows data from EasyBase to be quickly transfered onto another program. One application of this feature would be something like this; you're using Quill to write a letter to a friend. To find your friends address, you fire up EasyBase, load your address database, do a find on the friend, then copy the record into the Stuffer Buffer, switch back to Quill, hit ALT-Enter and the address is typed into Quill for you. This really saves time if you are writing a number of letters all in the same session.

Exporting Data out of EasyBase

EasyBase allows you to transfer data from EasyBase to other applications (be they databases, spreadsheets, etc.) by exporting the database in a couple of different formats. The most common format used between databases is the Comma Seperated Variable format (CSV). In this format there are commas between fields, and test strings have quote characters around them. EasyBase also creates export files compatible with Archive. You can also tinker around with EasyBase and get the data exported in a variety of formats, including just plain text.

Importing Data into EasyBase

The Import command allows you to bring in data from other sources (including Archive) into EasyBase. The main criteria in using Import is defining the field seperator. As mentioned above, the CSV format uses commas as the field seperator. Another option, if importing Unix text databases, is using semi-colons for field seperators.

EasyBase expects the first line of the import file to define the field names. If importing a text database with just data in it, I recommend using a text editor and creating the first line with the names of the fields. Make sure it's in the same format as the rest of the file (using the same field seperator).

Given the limitations of importing data, there might be some irregularities in importing some data. It's a good idea to go through each record and make sure that all the data is there and correct. Any little glitch in the imported data file will affect EasyBase.

Printing from EasyBase

The longest section in the manual covers printing from EasyBase. There is a lot to it. EasyBase gives a lot of flexibility is what can be sent to a printer (as it should). You can print by columns, rows, labels (as in mailing labels) and seperator (similar to exporting). You can select the fields to print, select certain criteria for a field (print only those fields with a certain value), from a certain range, and so on. This is one command that one really needs to read the manual on.

Another part of EasyBase important to printing is the configuration options. The Print command defines how you want to print this particular database (or view of the database), the print configuration options define how to print to your printer. You need to make sure that these are set properly before using the Print command. I highly advise taking your time in this process, going back to the manual to answer any questions that crop up during the process.

Summary

Personally, I'm a little surprized to see a new applicatoin program developed without using the Pointer Environment (PE). Just as MS-Windows took over for MS-DOS, I thought that the PE took over for plain QDOS. But, I really don't talk with many other QLers, so I don't know if there is a segment of QL users that are not using the PE (and may not want to use the PE). I'm sure that there is a hidden number of QL users that have yet to discover the Internet (where I do all of my communication with QLers) and don't write for the QL magazines. It could be this population where EasyBase is aimed.

Adding to Your QL

Dilwyn Jones

This article was inspired by correspondence I had with a former QL user who had rediscovered his QL ten years after consigning it to his attic in favour of a computer like the one he used at work. Following his retirement, with more time on his hands, he had stumbled across a QL-related Web site and found that there was still a lot of interest in the QL. Having sent a few emails and made a few telephone calls he dug out his unexpanded QL, found it still worked after ten years in storage and rediscovered an interest in writing SuperBASIC programs.

On the internet, he had found out about new hardware like the Q40 and various QL emulators, but was mostly interested in spending as little as possible to kit out his QL as a minimal system purely for SuperBASIC programming and a bit of word processing. During his correspondence with me, it became clear there was a lot of information available about recent developments, but not about products from intervening years. So I decided to write an article about expanding a QL based system. This article will double up as a sort of history of the QL and a teach-in for those returning to the QL after a period of absence and who wish to update their knowledge.

QL expansions mainly fall into the following categories:

- memory enhancements
- adding floppy disk systems
- toolkits
- printers
- combined expansion units
- operating system upgrade
- hard disk systems
- flash memory cards
- spare parts
- mouse systems
- keyboard upgrades
- new cases for the QL
- processor upgrades

Expansion Ports

The QL has several ports which allow other devices to be plugged into the QL, or other QLs to be connected to the QL.

1. SYSTEM EXPANSION SLOT

On the left hand side of the QL there is a removable cover which conceals a 64 way peripheral expansion slot. This is the main expansion route -

you can plug in memory cards, floppy disk expansions etc in here.

2. EPROM SLOT
Arund the back of the QL there is a 30 way expansion slot into which an EPROM carrier board may be plugged. This board can only carry a 16kilobyte EPROM and is most commonly used for plugging in extra system ROM, such as extensions for the SuperBASIC interpreter (so-called Toolkit ROMs, see below), although over the years, ingenious hardware designers have developed ways of using this slot to add extra peripherals to the QL, since the QL itself has only one expansion slot.

3. MICRODRIVE EXPANSION SLOT
This little slot on the right hand side of the QL is only for adding additional microdrives to the QL. As Sinclair never produced these add on microdrive units, this slot would appear to be redundant, although it can have two uses. Firstly, it carries a small amount of power, so add-ons could in theory draw some power from here, and secondly older Spectrum microdrives could be used, although the cabling means that the Spectrum microdrive will end up facing the wrong way! If one of your QL drives is not too reliable, you could for example copy microdrive cartridges by putting the original in the Spectrum microdrive (which will be MDV3_) and copy to the usable QL microdrive. One word of warning, the Spectrum microdrives will not always format cartridges for a QL successfully, although they will usually read QL cartridges OK. Although QL and Spectrum microdrive cartridges are physically identical, in practice the two computers format their tapes differently, the QL usually giving a slightly higher capacity.

4. NETWORK SOCKETS
These allow you to connect two or more QLs together with simple two wire leads to form an effective if rather slow network link between computers. Thus, more than one QL (up to 64 in theory) could share printers, for example. Unfortunately, only other QLs can be connected to this port - you could not use it to connect your QL to a PC or Apple Mac at work for example.

5. SERIAL PORTS
These are fairly standard RS232C ports, which can be used to connect the QL to printers, modems and even other computers with suitable cabling. In fact, there is a system called Sernet (developed by Bernd Reinhardt from the earlier Midinet) which allows a form of networking between the QL and other computers running QL compatible operating systems or QL emulators by using a suitable serial cable. The QL has two serial ports, but there are limitations on their use - the maximum speed is either 19,200 baud or 9,600 baud depending on what they are being used for, and it may be hard to use them independently for different applications.

6. JOYSTICK PORTS
These ports at the back of the QL have received little attention, but they do allow two Atari-style switched joysticks to be added to the QL. For simplicity, the joysticks pretend to be keypresses, the up/down/left/right joystick movements emulate the cursor key presses and the fire button pretends to be a SPACE keypress, making it easy to read the joystick with the INKEY$ function from SuperBASIC for example.

7. VIDEO
The QL has a television modulator, allowing the computer to be used with a domestic TV set. It also has a composite video output for connection to monochrome monitors or to some video recorder inputs. The same connector allows connection of older RGB video monitors, but not to modern SVGA monitors sadly. Articles in QL Today have shown how to connect the QL to SCART/Peritel connectors on modern TV sets.

Over the years, a number of different types of QL peripherals have been released. Next, we'll take a broad look at the various types of devices produced. Not all of them will still be available, although most have a habit of turning up second-hand at computer shows or via the traders who still manage to obtain and sell such QL equipment. A quick look through recent QL magazines such as QL Today and Quanta newsletter along with QL-related Web sites if you have internet access will give you addresses and contact details for traders still supplying QL products.

Memory Add-ons

The basic unexpanded QL comes with just 128 kilobytes (KB) of RAM memory. While this might be enough for writing simple BASIC programs and simple word processing, it is very restrictive for all but the most basic of uses of the QL. In theory, you can add extra memory inside the QL (in the 1980s some companies offered this as an upgrade) but it is more common to buy a memory expansion card which plugs into the system expansion slot on the left hand side of the QL. The original memory expansion cards tended to have only 128 or 256 kilobytes of memory on them, but as larger memory chips became more common and prices fell, manufacturers standardised on the maximum design limit for memory of a standard QL of 640 kilobytes (128KB standard QL memory plus 512KB on an expansion card). For many years, this was the 'standard' expansion memory for a QL and indeed these units are still quite adequate if all you want is to add some memory to the QL and are still quite happy with a microdrive based system. The problem with these memory expansions is that many of them have no through connector to allow connection of other peripherals such as disk interfaces (a "through connector" is a duplicate of the 64 way expansion connector inside the QL, in effect allowing you to 'daisy chain' another expansion card). If you think you are likely to want to buy other devices such as floppy disk or hard disk interfaces, it is well worth paying a little extra to get a memory expander with such a through connector. Examples of suitable 512KB expansion cards were the Miracle Expandaram, Sandy 512K expansion, Silicon Express Insider and the CST 512KB ram card.

Note that you can only add one memory expansion card. If you buy an old 256KB unit, and then decide to try to add another 256KB, this won't work.

Adding memory to your QL system does not increase operating speed of the computer to any significant degree, although a 128KB memory QL without floppy disks may show improved microdrive performance through the use of the extra memory to improve buffering of the microdrive cartridge data by holding copies of the data in memory (called 'slave blocks').

In practice, this type of memory-only card was superceded by expansion units also containing floppy disk interface systems and these days both types of units cost very little where they can still be obtained.

Floppy Disk Systems

A floppy disk system will greatly improve your QL in many respects. Floppy disks hold much more information than microdrive cartridges, are usually cheaper, much more reliable and faster. Microdrive cartridges are no longer being produced, so although there are plenty of second hand cartridges available, few QL users now struggle along without floppy disk systems.

The most popular floppy disk systems adhere to the so-called QJump floppy disk format these days, rather than the original 'official' Sinclair systems, although this shouldn't be a problem as the original Sinclair/Micro Peripherals disk systems are now hard to obtain.

Most QLers use the standard 3.5 inch floppy disks as used by most computers. Most interfaces can also handle the old 5.25 inch disks and also the sturdier but less common and more expensive 3 inch disks used by companies such as Amstrad in their older computer systems. Few (if any) QL software suppliers will now supply software on anything other than 3.5 inch disks so buying anything else could be a false investment.

The older QL floppy disk interfaces will only handle Double Density (720KB capacity) floppy disks. These are going to become increasingly scarce in the future as manufacturers concentrate on the High Density disks (HD disks have a capacity of about 1.4 megabytes), although the QL community still commonly uses DD disks. More recent disk interfaces such as those on the Gold Card and Super Gold Card expansion units (more on these later) can handle HD disk drives and even the rarer ED or Extra Density 3.2 megabyte disks. Note that these 3.2MB disks are pretty non-standard, as the QL is the only computer I know of to use them - the only other computer I have seen to use them formats these disks to 2.8 MB instead.

One useful feature of QL disk systems is that HD and ED drives can read from and write to lower density disks than that they were designed for. So HD disk drives can read and write to DD disks as well, while ED disk drives can also use DD and HD disks. The reliability of using lower density disks is not as good as using the disks for which the drive was designed, however. It is not unusual to find that a DD disk you have copied on your ED disk drive fails to read properly on a friend's DD disk drive, for example.

Expansions which contain only disk interfaces are usually quite suitable for adding to a system containing a memory expansion, provided that there is a means to plug both into the QL simultaneously. As many memory cards include a through connector, this is not necessarily a problem. There are also 'backplane' units such as the QPlane from Qubbesoft and an unit called MPlane from TF Services which add extra expansion slots to the QL allowing extra cards to be plugged in.

Disk interfaces for the QL were produced by several companies, including CST, Miracle Systems, Silicon Express, Sandy and PCML among others. I used a CST QDisk with a Miracle Expandaram card quite happily for a few years.

Some of these disk interfaces contain a subset of the toolkit extensions mentioned (see Toolkit 2 information later on). A few contain the full Toolkit 2. When looking for a suitable disk interface, ask if it includes Toolkit 2 as it will save you having to buy a plug in Toolkit 2 later.

Disk interfaces can be used alone without any memory expansion, but as the disk interface system needs a little bit of memory to itself, it reduces even further the meagre 128KB of memory on a standard QL, so the use of just a floppy disk system without adding any memory is of dubious benefit unless you are just going to write your own small SuperBASIC programs or use Quill to produce short documents, for example.

After a while, manufacturers stopped producing these stand-alone disk interfaces and memory cards and moved towards all-in-one designs.

Toolkits

Not hammers and spanners, but add-on software which adds facilities to either the operating system or to SuperBASIC. The most common one is called Toolkit 2, by Tony Tebby, one of the original designers of the QL's operating system. This was an improved version of the original QL Toolkit released by Sinclair's software division in the early days of the QL. Toolkit 2 (also called Super Toolkit 2 in some cases) adds many new facilites, including an almost essential set of new commands and functions for SuperBASIC. These have become standard features on many users' QL systems over the years and I for one find the facilities offered by Toolkit 2 to be indispensable.

Toolkit 2 has appeared in three main forms over the years. The first was a plug in EPROM cartridge, a small plastic cased board carrying a 16KB EPROM chip which plugged into the EPROM expansion slot at the back of the QL. This was a very useful system, as it meant that the Toolkit was instantly available each time you switched on the QL without having to remember to load the toolkit software from disk, for example. A second version gave you Toolkit 2 on a floppy disk or microdrive cartridge, allowing you to select which sections of the toolkit software you wanted, to save space by reducing the size of the toolkit by removing unwanted facilities. This was called the Reconfigurable Toolkit 2. While useful in some respects, this version of the toolkit was always less popular and harder to obtain than the EPROM version.

The third version of Toolkit 2 was the version built into many disk interface systems, especially those produced by Miracle Systems. If you obtain a disk interface with Toolkit 2 on board, you do not need to purchase it separately.

Combined Units

Several companies went on to integrate memory expansion units, disk interfaces and Toolkit 2. Examples of such units were the Super-Q-Board from Sandy, a similar unit from PCML (both those took the QL's RAM to a total of 640KB) and a more ambitious device called the Trump Card from Miracle Systems. This unit rather broke the rules by taking the total RAM to a then massive 896KB by making use of space designated by the QL designers for peripheral expansion to provide extra RAM over and above the QL's theoretical design limit of 640KB. The downside of this massive memory though was that no further peripheral expansion was possible as all designated memory areas were now fully used up. A second version of the Trump Card was later released by Qubbesoft - both the Miracle and Qubbesoft versions are ideal low cost expansion units to resurrect an old QL, or to equip a spare or backup QL for your main system. Sadly, none of these older combination units are produced any more, so unless you strike lucky and stumble across one being sold second hand somewhere, or a trader gets one in for sale, you'll be lucky to get hold of one.

Most of these older systems could only handle one or two floppy disk drives, and only DD drives at that.

Miracle Systems later produced a combination system called the Gold Card, which included a disk system capable of handling DD, HD, or ED drives and either 2 or 4 drives depending on whether you bought a 'drive expander' to allow it to handle drives 3 and 4 or not. The Gold Card gave you a massive 2MB of RAM memory, Toolkit 2 and even a new microprocessor to replace the 68008 chip in your QL. The 68000 chip in the Gold Card allowed your software to run many times faster than the original QL and its success was assured as a result. While Gold Cards are no longer made, they are fairly common second hand and make ideal expansion units for those looking to expand their basic QL system.

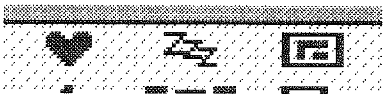
Miracle Systems went one step further and produced an even better version of the Gold Card, imaginatively called the Super Gold Card. This gave an even bigger 4MB of RAM memory along with an even faster microprocessor, a version of the 68020 chip. In addition, a Centronics-compatible parallel printer connection was included on the board along with the facility to use up to 4 disk drives without a need for any external adaptors. After being discontinued by Miracle Systems, the Super Gold Card was first produced by Quanta and then by QBranch. Component shortages brought its production to a slow end fairly recently, but there are plenty of units out there and they often become available second hand - QBranch often have a few Gold Cards and Super Gold Cards available for sale, for example.

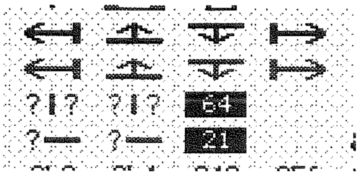
In the next issue we'll look at operating system upgrades and hard disk systems.

About Sprited

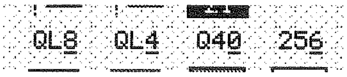
Jérôme Grimbert

As the author of sprted reviewed in issue 3, I would like to give some precisions to the readers of QL-Today. Since the 25 october 2000 (and not before), the sprite editor program can be used on QPC2 v2 (the prerelease with High colour) and QXL (also with high colour). In fact any system with a display big enough and a GD2 driver should be ok. Previously, running sprted (the sprite editor) was only possible on a Q40 (also with high colour). Nevertheless, when not on a Q40, you cannot have the Q40 specific sprite (Just tell me how to display 65535 accurently on a 256 colour system). I'm sorry for the other systems (such as a standard QL), but I probably won't make it possible to run on a mode 4 only (yet, you can design a mode 4 only sprite on the supported systems). I would like also to correct the reviewer about saving capability: there is no output in pic format. Only assembler is possible. It is possible to load a pic as a mask and this button was near the 'save asm' one, so I can understand the misunderstanding. Let's now have a guided tour of the hundreds of buttons in sprted, from the top to bottom at the center of the main window. The heart is my 'Quit' buttons, then there is the 'sleep' and then the resize which does currently nothing because I found the window is already large enough. Still on the classical PE buttons, we have the 'Wake' which redraws the windows, the 'change mode' which is unavailable and the 'Move' which allows you to put the big window where you want on a big resolution system.

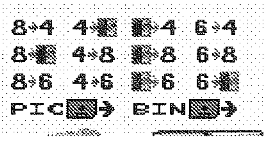




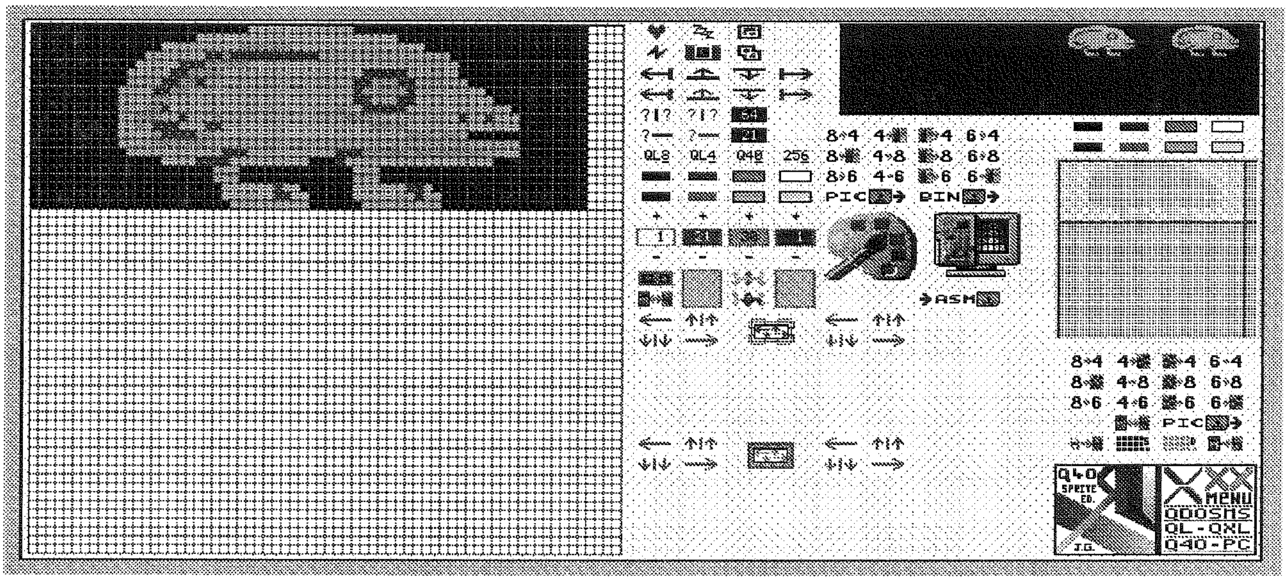
The sizing control. The current sprite is delimited by two set of parallel lines, one red and one black, which define a rectangular area. You can either move a line by one step (usually one pixel, excepted in mode 8 where the horizontal move is two pixel at a time, due to the implementation of the mode 8 on the original QL [remember, in mode 8, the 512x256 pixels of mode 4 get seen and addressed as only 256x256 pixels]) with the first two lines of buttons, or directly set the position with the last two lines. Also on the last two lines, there is the current wide and height of the sprite.



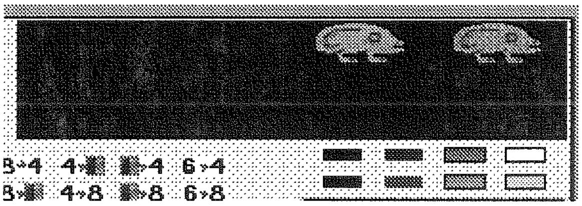
On this line, you can choose which sprite version you want to edit. When choosing a new mode, the previous size-lines are automatically copied in the new mode. So on a Q40, you can choose between mode 8, mode 4, mode 33 (the Q40 native mode) and the 256 fixed colours (mode 16). I do not really see the interest of having a palatted sprite, so mode 31 is not supported (even if it is supported by all GD2 compatible display). On a non-Q40 system, as already said, the Q40 native mode is unavailable.



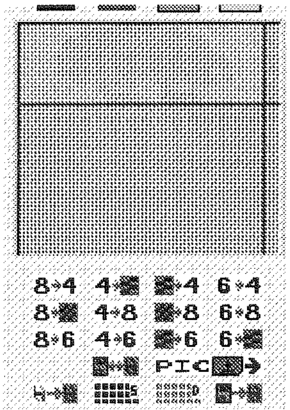
On the right of the main column of flashy buttons, lies the buttons which deal with the automated conversion between the various modes of the pattern (the part of the sprite which gives the colours informations). The Q40 mode is represented as a colourful checker, mode 4 and 8 by respectively 4 and 8, and the lately arrived 256 colours by a 6 (because 256 ends with a 6 !). The first colums has all buttons which convert from



mode 8, the second from mode 4, the third from Q40 and the fourth from the 256 colours mode. Similarly, the first row holds all the buttons that have to deals with mode 4, the second row with mode 8 and the third row about the 256 colours mode. The disposition has therefore some kind of logic even if a true matrix of conversion would have required a 4 x 4 grid with an empty diagonal (no point to convert from mode 4 to mode 4 !). The last two big buttons at the bottom allow to read a _pic for the pattern (either in mode 4, mode 8 and mode 33, I have not yet made the loader for a pic taken under QXL or QPC2 in high colour mode) and to explore a binary file (such as a program file) looking for some possible sprites (this scanner works for the four possible modes).



On the top right corner, we have the preview of the sprites. The order from the left to the right is similar to the mode selection buttons: mode 8, mode 4, Q40 and 256 colours. There is under the preview a quick choice from the eight basic colours to use as the background colour for the preview. Beware nevertheless, the mode 8 preview is emulated as a mode 256, so the display may not be accurate, and handling of mode 4 may differ according to the underlying hardware and software (especially when it is a true mode 4 hardware, there is no such thing as yellow, cyan, blue or magenta). The main interest of being able to change the background colour is to check for visibility and readability of your sprite (the cute one on white may be very hard to understand when on black or red).



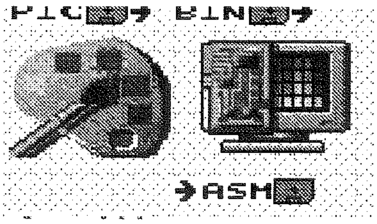
In the middle of the right side, there is the current mask: blue for opaque pixels and light green for transparent pixels. If you HIT on the area, the mask is transfered into the main edition window so you can edit it, and if you DO again, the main edition window comes back to editing the pattern. Under the mask window, we find again a set of buttons to perform automatic transfers of the mask between the various modes (same logic than with the pattern transfer).

QBOX-USA BBS

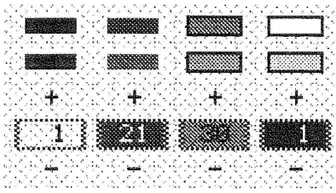
Operating since 1993 on a Sinclair QL from Utica, Michigan, USA
Supporting *ALL* Sinclair and Timex users
Message and File Areas for QL, Z88, Spectrum, TS2068, ZX81, TS1000
Modem speeds 300 bps to 33.6k bps supported
24 hour operation - call anytime
810-254-9878

Then we have a small button to inverse the mask (transparent become opaque and vice-versa) as well as a button to load a pic as a mask (black is transparent, other is opaque). On the last line, from the left to the right, there is:

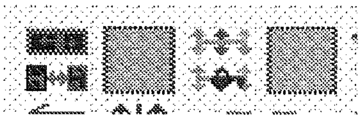
- a button to compute the mask from the pattern, where bright white is transparent and others is opaque.
- a button to set the whole mask opaque (keyboard shortcut is S)
- a button to set the whole mask transparent (keyboard shortcut is D)
- a button to compute the mask from the pattern, where black is transparent and others is opaque.



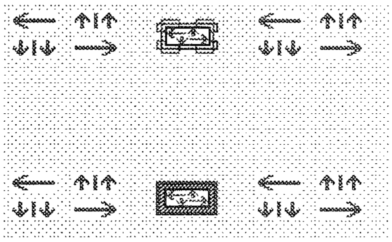
Going back to the middle of the main window, we encounter two really big buttons which allow you to choose the colour for the pattern edition. The one with the brush is only available on Q40, the other one (with the computer monitor) displays the choice of 256 colours. Under them, there is the 'Save to assembly' button. You can choose if you want to generate assembly for QMAC or AS68 via a config block. You can choose which modes of the sprite to save, as well as the name of the file to write the sprite to and the name that will have the sprite when used in C. If the designated file already exist, you are prompted to check if you want erase it with the new sprite.



Right above the very center of the main window are the buttons to quickly select a basic colour and to adjust the White/Red/Green/Blue components, as well as the values of each components. The fine adjustment is unavailable when working with a mode 4 or a mode 8 sprite.



Then we have on the leftmost part a button to fill the current sprite with the current colour (which is displayed next to it), as well as a button to inverse the colours of the current pattern. On the right of the preview of the current color, there are two buttons which deal with a secondary colour: the top button exchange the current colour with the secondary one (whose preview is on the right), while the other button will scan and replace in the current pattern the secondary colour with the current one (very useful to replace a shade of red with another shade of blue, without having to do it by hand & eyes).



And finally, some basic manipulations of the pattern and the mask. On the left, the manipulation applies to the pattern and on the right to the mask. The top rows make a rotation, the part which goes out of the sprite is put back at the other extremity, while the bottom rows perform a translation: the part which goes out of the sprite is lost and the other extremity get the current colour. For each manipulation, you can have any of the four directions (up, down, left, right) and it is one pixel at a time. So this end the guided tour of the buttons. A last word about the mouse click on the main editing window. When in pattern mode, a HIT puts the current colour in the cell under the pointer (you can keep HIT down and move the mouse, it should paint the cells which are moved upon [you may have some gaps due to your mouse settings]), and a DO takes the colour of the cell under the pointer as the current colour. When in mask mode, HIT will set the pixel opaque and DO will make it transparent.

Last recall: sprted is available as freeware from **http://grimbert.cjb.net/**
(this is a redirection, but it can be bookmarked) and **http://www.crosswinds.net/~grimbert/**
(this is the actual hosting server, but it may change, so beware)

Please note that the features described in the previous and next article can be used only on QXL with Hi Colour driver upgrade, qpc2 Version 2 and Q40s! At present, only these system can handle more than the QL MODE 4 and 8 colours!

TurboPTR and SPRITES/ BLOBS/PATTERNS on the Q40

George Gwilt

A Problem

TurboPTR is a set of programs enabling pointer applications to be written in SuperBASIC and then compiled by TURBO.

One of the programs comprising TurboPTR is TPTR_SETF_TASK (SETF for short) which helps in the designing of windows for an application. Each main window and each window with a menu must have a sprite to act as the pointer. To help the programmer make his choice, each of the sprites available, as well as its name, can be displayed one at a time by SETF. By using the up and down arrows you can scroll through the list of sprites available.

Imagine my surprise when one day while I was using SETF on my Q40 in extended colour mode I found that the displayed sprite remained the same, although its name was changing as I scrolled up and down through the list. This certainly indicated a F... in the SMSQ/E software! However, F... turned out to be FEATURE, not FAULT as I first thought. Let me explain.

Each time a sprite is displayed by SETF the file containing it is copied to RAM into an area allocated from the heap. The sprites in my list were defined only for Mode 4. The SMSQE software allows for this in the Q40 when the extended colours are in operation via Mode 33. The Mode 4 version is translated to a Mode 33 version. Since this is a comparatively lengthy procedure, to save time the result is stored in a special buffer and used again and again so long as the sprite's address in RAM has not changed. Since SETF, in going through the list of sprites, was continually loading these different sprites to an area allocated from the heap, then releasing that area ready for the next sprite, it was no wonder that, sometimes at least, the same address was used for the different sprites, thus resulting in no change in the displayed sprite.

This feature is not confined to sprites. It is perfectly in order to use blobs and patterns as loose menu items in a main window and as items in information windows. If you choose a blob as one of the entities in question, SETF allows you to scroll through the available patterns so that you can choose the one you want. These patterns refused to change too. And for the same reason.

What are Blobs and Patters?

Sprites are familiar to users as the objects which move around the screen when a mouse is moved. But what are blobs and patterns? The short answer is that a blob is a shape without colour and a pattern is a colour without shape, whereas sprites have both shape and colour. So a sprite is just a blob combined with a pattern? Not quite!

A sprite is a rectangular array of pixels each with two attributes, colour and transparency. The transparency indicates whether the pixel is to be written to the screen (1) or whether it is to be XORd to the screen (0). A black pixel with transparency 0 will have no effect on the screen since black has the value zero and zero XORd with any number leaves it unchanged so the pixel on the screen remains unchanged. Thus pixels in the sprite which are black and have transparency 0 are effectively transparent. The set of colours for each pixel is called the colour pattern and the set of transparencies is called the pattern mask.

A sprite as it appears inside the computer for use by the Pointer Environment (PE) has a header indicating what Mode it is, its time constant, what size it is, what its origin is, where the colour pattern is, where the pattern mask is and where the next sprite in the chain is.

The size is the width and height in pixels. The origin is that pixel in the sprite which is the true point. If you move a sprite into an area occupied by a menu item, for example the move icon appearing in almost every PE program, you will notice that a border appears. Close examination will tell you where the "origin" of the sprite is. For example the origin of the arrow used in QD is, as you might expect, at the arrow's tip. The sprite's size, as already mentioned, is a rectangle. The origin of the sprite is measured from the top left corner of this rectangle. Positive x and y values place the origin to the right and below the top left.

A blob has a header the same as a sprite, though the time constant and the pointer to the colour pattern are both zero. To be of any use a blob MUST be associated with a pattern.

A pattern can be thought of as an indication of what will be found "underneath" the screen when the associated blob is pulled away. The pattern defines the colour of each pixel in an area which is deemed to be repeated over the entire window. Probably the commonest form of pattern is simply a solid colour. A blob associated with such a pattern would appear on the screen in that colour.

As an example, to define a green pattern covering the whole window all we need is a size of eight pixels wide by one in depth. The colour mask is then just eight pixels of green.

A Solution

The problem stated above would disappear if the sprites/blobs/patterns were defined for the correct Mode, which is 33 in the case of the Q40. This is because there is no need to translate the supplied sprite if it is already in the required format. The software to print the sprite would then use the address of the sprite itself and not the address of the special buffer used for translations.

To allow for different Modes, sprites can be linked together. The software will look through this chain to find the appropriate item with the desired Mode. If the Mode is 8 but there is no Mode 8 sprite in the chain the default sprite will appear. This does not apply with the extended colour Modes. To allow existing programs to operate with the new Modes, the software arranges to translate the sprites to the "native" colour Mode if there is not a "native" sprite defined in the chain.

Such a linking of different Modes applies also to blobs and patterns.

Dynamic Sprites

I mentioned earlier that in the heading for a sprite there was a time constant. This can be used to produce sprites which alter with time. To achieve this the set of sprites to be displayed must be linked together in such a way that the time byte increases from the earliest to the latest.

If you now want to define a set of dynamic sprites for different Modes, you have to link the dynamic chain for one Mode to the dynamic chain for the next Mode and so on.

If, for example, you have a dynamic sprite of four elements sp1_m, sp2_m, sp3_m and sp4_m for Mode m, the complete chain for Modes 4, 8 and 33 would be:

sp1_4, sp2_4, sp3_4, sp4_4, sp1_8, sp2_8,
sp3_8, sp4_8, sp1_33, sp2_33, sp3_33, sp4_33.

Although you can't have dynamic blobs and patterns, you can chain them as you can sprites for differing Modes.

One way of doing this is to use the program CHSPR (for CHain SPRites), supplied with Turbo-PTR. This allows the chaining of up to six entities, all of the same type and permits sprites to have a timing byte added. To arrive at the sp1_4 etc chain mentioned above you would first apply CHSPR to the four sprites Mode 4, then the four sprites for Mode 8 and finally the four sprites for Mode 33. Then the resulting three sprites would be linked by CHSPR.

Production of Sprites

All this is very fine if we have a set of sprites in the first place. When a sprite is being used by a PE program, it has been loaded into RAM and is accessed by a pointer to its address. But how do we get it into RAM?

The most obvious way is to have a file containing the binary representation of the sprite and simply load it directly into RAM by LBYTES in Super-BASIC or FS_LOAD in assembler. In TurbPTR there is a function RD_SPRT(o%) which returns the address of an area in RAM where a sprite (or blob or pattern) has been set up from information in the next DATA statement to be READ. This can then be saved (by SBYTES) to a file for later use. The format of the DATA statements needed is given later.

The value of o% determines how RD_SPRT should operate. If o% is 0, an ordinary sprite is produced. If o% is 2, the sprite is "transparent". That is, it is set so that all of its pixels are XORd into the screen, just as for the "zzz" sleep sprite. For a blob, o% is 1 and for a pattern -1.

At the moment RD_SPRITE is not capable of producing sprites, blobs or patterns for Modes other than 4, 8 and 33.

Other programs to produce sprites are Super-BASIC procedure SPSET and program EDSPR_BAS, both from QPTR, and Jerome Grimbert's C68 program, SPRTED, which was the subject of an article by Bruce Nicholls in a recent edition of QL Today (Volume 5 Issue 3).

The procedure SPSET is similar to RD_SPRITE and sets a sprite image in RAM. EDSPR_BAS and SPRTED both produce assembler code as output. In the case of EDSPR the sprite image is first built up in RAM by SPSET and then translated into assembler format for output. The output from SPRTED can be either for QMAC or AS86 (evidently for C68). Happily the output for QMAC can be assembled by GWASS (also available for C68). However, it seems slightly annoying to have to assemble output back to the RAM image from which the assembler output came to arrive at the direct binary representation needed for TurboPTR.

The reason for the assembler output is that in QPTR the method adopted for writing PE programs by assembler was to produce a set of linkable modules, of which one contained the section SPRITE. In C68, too, the method of compilation expects a section called SPRITE. Although it is straightforward to produce either the binary output of RD_SPRT or the assembler code of SPRTED or EDSPR_BAS, it is less easy to produce directly as output the assembled relocatable _o file needed for linking. Hence the reliance on an assembler to produce this. However, assemblers are well able to incorporate binary directly into their output and I suggest that it would be very nearly as easy for an assembler to produce the relocatable output by use of the direct binary output as from the assembler output. For example, if you are using the assembler GWASS and want two sprites called ima1 and ima2 in your program you will need:

```
      XREF   ima1,ima2
```

in the program's MAIN section. This will enable pointers to the sprites to be set up in MAIN.
For the SPRITE section you would assemble together the two sprite outputs called, let us say, ram1_spima1_ASM and ram1_spima2_ASM. This needs:

```
      IN      ram1_spima1_ASM  The files are
                         INcluded in the . .
      IN      ram1_spima2_ASM   . . instructions to
                         be assembled
```

To use the binary files instead called, say, bima1 and bima2, we need:

```
      SECTION SPRITE         These are part of . .
      XDEF    ima1,ima2      . . "ram1_spima1/2_ASM"
ima1  LIB     ram1_bima1_BIN The files are added . .
ima2  LIB     ram1_bima1_BIN . .to the binary output
```

This is fairly simple. With the addition of a macro this can be made even simpler.
Given the macro BSPR:

```
BSPR  MACRO   filename,spritename
      SECTION SPRITE
      XDEF    \2           (This sets the sprite name)
\2    LIB     \1_BIN       (This sets the filename)
      ENDM
```

we need only write:

```
      BSPR    ram1_bima1,ima1
      BSPR    ram1_bima2,ima2
```

to produce the correct SPRITE section.

Perhaps Jerome Grimbert could be persuaded to add binary output to SPRTED.

FORMAT of DATA STATEMENTS for RD_SPRITE

Bruce Nicholls in his article on SPRTED referred to 'hand coding' of the sprite as the alternative to using either EDSPR or SPRTED to draw the sprite on the screen. In fact both EDSPR and RD_SPRITE use very similar DATA statements to define the sprite. Although the sprite does not appear on the screen while the DATA statements are being typed, nevertheless the format of the sprite becomes clearly visible. This can be seen in the following example, which includes the instructions necessary to produce the file containing the sprite.
The function SPR_SZE(x,y) returns the amount of space needed for the sprite. For a sprite of Mode 33, x muxt be entered negatively. As explained above, the width and depth of the sprite are recorded in bytes 4 to 7 so can be found by PEEKing.

```
100 RESTORE 1160:k=RD_SPRT(0)
110 SBYTES ram1_tree_spr,k,SPR_SZE(PEEK_W(k+4),-PEEK_W(k+6))
120 RECHP k:STOP
130 :
140 :
150 :
1000 REMark ************************************************************
1010 REMark * The first three items are the origin and the Mode.      *
1020 REMark * The subsequent lines are successive lines of the sprite. *
1030 REMark * SPACE indicates a transparent pixel.   For Mode 33       *
1040 REMark * all other pixels are given a colour represented by a    *
1050 REMark * different character.                                     *
1060 REMark *     w = white                                            *
1070 REMark *     g = green                                            *
1080 REMark *     l = mauve                                            *
1090 REMark *     f = shocking pink                                    *
1100 REMark *     6 = dark red                                         *
1110 REMark *                                                          *
1120 REMark * If TPTR_BAS is MERGEd with this, RUN will produce the    *
1130 REMark * sprite "ram1_tree_spr". The extras TPTR must be loaded. *
1140 REMark ************************************************************
1150 :
```

```
1160 DATA 13,0,33
1170 DATA '             w'
1180 DATA '            wgw'
1190 DATA '           wgggw'
1200 DATA '          wgggggw'
1210 DATA '         wglgggggw'
1220 DATA '        wgglggggggw'
1230 DATA '       wggfffggggggw'
1240 DATA '      wgggfffgggggggw'
1250 DATA '     wwwwgfffgggggwwww'
1260 DATA '       wggfffggggggw'
1270 DATA '      wgggggggggglggw'
1280 DATA '     wggggggggggglgggw'
1290 DATA '    wgggggggggggfffgggw'
1300 DATA '   wggggggggggggfffggggw'
1310 DATA '  wgggggggggggggfffgggggw'
1320 DATA ' wggggggggggggggfffggggggw'
1330 DATA 'wwwwwwwwwwwwwwwwwwwwwwwwwww'
1340 DATA '           w6666w'
1350 DATA '           w6666w'
1360 DATA '           w6666w'
1370 DATA '           w6666w'
1380 DATA '           w6666w'
1390 DATA '           wwwwww',''
1400 :
```

QLTdis - part 5

Norman Dunbar

As promised at the end of part 4, we will have a detailed look at the sub-routines now:

The following are subroutines called from various parts of the types decoding routines explained in the previous issue.

Condition code routine

This routine is called when we have a condition code to decode.

INPUT:
D0.B = condition code.

OUTPUT:
Nothing. Adds the condition code from the table below to the output buffer and updates A5 and D6 appropriately.

PRESERVES:
Preserves all registers that matter.

The first two conditions are only one byte each. Treat separately.

```
If D0.B = 0
   add 'T' to the output buffer
   exit sub-routine.
end if
If D0.B = 1
   add 'F' to the output buffer
   exit sub-routine.
end if
```

All other condition codes are two bytes. So create a table of them all in numeric order. See CC_TABLE below.

```
Save D0 and A3 on the stack.
Point A3 at CC_TABLE.
Shift D0.B 1 bit left (D0.B = D0.B * 2)
Get the word at A3.L + D0.B into D4.W
Add D4.W to the output buffer.
Restore D0 and A3 from the stack.
Done.
```

```
CC_TABLE DC.W    "T "    0 = T => Special case
                         - never read from here
         DC.W    "F "    1 = F => Special case
                         - never read from here
         DC.W    "HI"    2 = HI
         DC.W    "LS"    3 = LS
         DC.W    "CC"    4 = CC
         DC.W    "CS"    5 = CS
         DC.W    "NE"    6 = NE
         DC.W    "EQ"    7 = EQ
         DC.W    "VC"    8 = VC
         DC.W    "VS"    9 = VS
         DC.W    "PL"    10 = PL
         DC.W    "MI"    11 = MI
         DC.W    "GE"    12 = GE
         DC.W    "LT"    13 = LT
         DC.W    "GT"    14 = GT
         DC.W    "LE"    15 = LE
```

QUANTA

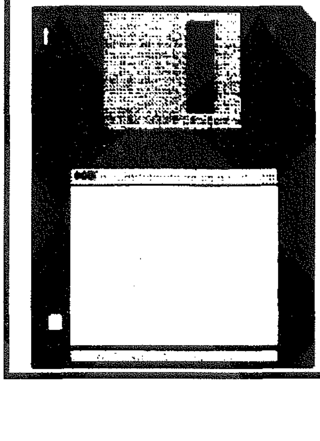
Independent QL Users Group

Worldwide Membership is by subscription
only, and offers the following benefits:
Monthly Newsletter - up to 40 pages
Massive Software Library - All Free !
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software and Hardware problems
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our SEVENTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407

Size decoding routine

INPUT:
D0.W = op_code - we want bits 6 and 7.

OUTPUT:
Sets D5.W to correct value for Effective address routine (1, 2 or 4).
Adds the '.B', '.W' or '.L' to the output buffer followed by a space.

PRESERVES:
D0.W

CORRUPTS:
A3, D5 and D4.

```
Copy D0.W to D5.W
Call the 'size into D0' routine. (Corrupts D0)
Exchange D0 and D5
Point A3 at S_TABLE (See below)
Get the word at A3.L + D5.W into D4.W and add it to the buffer.
Move a space into D4.B.
Add it to the buffer.
Done.
```

```
S_TABLE DC.W '.B'
        DC.W '.W'
        DC.W '.L'
```

Source register routine

INPUT:
D0.W = op-code - we want bits 0 to 2.

OUTPUT:
Adds the source register (0 - 7) to the op-code buffer and adjusts A5 and D5 as
required.

PRESERVES:
D0.W

```
Copy D0.W to D4.W.
Mask out all but bits 0 to 2 of D4.W.
Convert D4.B to hex and add to op-code buffer.
Done.
```

Dest register routine

INPUT:
D0.W = op-code - we want bits 9 to 11.

OUTPUT:
Adds the dest register (0 - 7) to the op-code buffer and adjusts A5 and D5 as
required.

PRESERVES:
Preserves D0.W

```
Copy D0.W to D4.W.
Mask out all but bits 9 to 11 of D4.W.
Shift D4.W right by 9 bits.
Convert D4.B to hex and add to op-code buffer.
Done.
```

Bit-op routine

INPUT:
D0.W = op_code, we want bits 5 & 6.

CORRUPTS:
D0, D4 & A3.

```
Mask off all but bits 6 and 7 of D0.
Shift D0.W 4 bits right. (4 is correct, = 0, 4, 8 or 12)
Point A3 at the BITS_TABLE.
Load the long word at 0(A3,D0.W) into D4.L
Add D4.L to the op-code buffer.
Done.
```

```
BITS_TABLE  DC.L   'TST ' Note trailing space!
            DC.L   'CHG '
            DC.L   'CLR '
            DC.L   'SET '
```

Size in D0 routine

INPUT:
D0.W = op-code, we want bits 6 & 7 only.

OUTPUT:
D0.W = old bits 6 & 7 shifted right by 6 bits.

CORRUPTS:
D0.W

```
Mask off all but bits 6 and 7 of D0.
Shift D0.W 6 bits right.
If D0.W = 0
   Move 1 to D0.W
end if.
Done.
```

Address register list routine

INPUT:
D4.B holds a bit map representing each address register. Bit 0 = A0 etc.

OUTPUT:
Adds the register list to the output buffer and seta A5 and D5 accordingly.

PRESERVES:
D0 to D4.

```
if D4.B = 0
   Done.
end if
Save D4, D3, D2, D1 and D0 on the stack.
Set D3 to 'A'
Call register list routine.
Restore D0 to D4 from the stack.
Done.
```

Data register list routine

INPUT:
D4.B holds a bit map representing each data register. Bit 0 = D0, etc.

OUTPUT:
Adds the register list to the output buffer and sets A5 and D5 accordingly.

PRESERVES:
D0 to D4.

```
if D4.B = 0
   Done.
end if
Save D4, D3, D2, D1 and D0 on the stack.
Set D3 to 'D'
Call register list routine.
Restore D0 to D4 from the stack.
Done.
```

Register list routine

INPUT:
D0.B = register list bitmap
D3.B = 'D' or 'A' according to the register type required

OUTPUT:
Adds the register list to the op-code buffer and sets D5 and A5 correctly.

I found this next bit of code to be the most difficult to explain in English because there is so much going on in such a short place. Basically we start scanning bits from bit 7 to bit 0 and when we find a set bit, we add that register name & number to the output buffer.
If the next bit is set, scan until we get to a clear bit and build the correct register list (Rn-Rm) in the output buffer. Otherwise we need to add a '/' and then look for the next set bit.
If we get to the end then we have a buffer which has one character too many in it and this is removed before we return from the sub-routine.

Watch out for this code next time we come back to QLTdis - it is a mind bender.

I have managed to convert the SuperBasic routine published by Andrew Pennell in his book 'Assembly Language on the QL' into assembly language. If anyone can put into words exactly what is going on here then please let me know.

For full details, check out the code in the next exciting (?) instalment of QLTdis and see if you can put into words what is going on!

Reverse bits in D4.W routine

This routine simply reverses the order of the bits in D4. Bit 15 becomes bit 0, bit 14 -> bit 1 and so on.

Effective address routine

INPUT:
D0.W = op-code - we want bits 0 to 5 (and maybe 6 and 7 if D5 = 0).
D5.W = size (1 = .B, 2 = .W or 4 = .L) or 0 = Decode size from bits 6 & 7 of D0.W

OUTPUT:
Adds effective address to the output buffer and sets D5 and A5 correctly.

PRESERVES:
D0.W

```
Save D0.W on stack.
if D5.W = 0
   Copy D0.W to D5.W
   Call 'size into D0' routine.
   Exchange D0 and D5.
end if
Clear D1.L  it will be the register number.
Copy D0.W to D1.W.
```

Mask out all but bits 3 to 5 of D0.W. (The mode)
Shift D0.W 2 bits right so mode 1 now = 2, mode 2 = 4 etc.
Mask out all but bits 0 to 2 of D1.W (The register).
Point A3 at mode_table
Get the word at A3.L + D0.W into D2.W.
Add D2.W to A3.L to get the address of the mode sub-routine.
Jump to the appropriate mode routine. (Return to here so use JSR not JMP)
Restore D0.W from stack
Done.

```
mode_table  dc.w    mode_0-mode_table
            dc.w    mode_1-mode_table;
            dc.w    mode_2-mode_table;
            dc.w    mode_3-mode_table;
            dc.w    mode_4-mode_table;
            dc.w    mode_5-mode_table;
            dc.w    mode_6-mode_table;
            dc.w    mode_7-mode_table;
```

There is no need to test for 'off the end of the table' values in D0.W because it can only have values between 0 and 7 (Doubled up to 0 to 14).

Effective address mode routines

The following are the mode routines for the effective address decoding. On entry ALL of them will have the following inputs:

D0.W = mode * 2
D1.W = Register number
D2.W = Don't care
D3.W = Don't care
D4.W = Don't care
D5.W = Size of instruction (1, 2, 4 for byte, word, long)
D6.W = Length of text in op-code buffer (so far)
D7.W = op-code word.

A0.L = Don't care
A1.L = Don't care
A2.L = Don't care
A3.L = Don't care
A4.L = Our programs dataspace pointer
A5.L = op-code buffer pointer
A6.L = Next word in memory after op-code word
A7.L = Stack pointer

There is no need to preserve D0 or D1 but the rest must be correctly updated.

Mode 0 (Data register direct)

Add 'D' to op-code buffer.
Convert D1.W to hex and add to the buffer.
Done.

Mode 1 (Address register direct)

Add 'A' to op-code buffer.
Convert D1.W to hex and add to the buffer.
Done.

Mode 2 (Address register indirect)

Add '(A' to the op-code buffer.
Convert D1.W to hex and add to the buffer.
Add ')' to the op-code buffer.
Done.

Mode 3 (Address register indirect with post-increment)

Call Mode 2 processing to add '(An)' to the buffer.
Add '+' to the op-code buffer.
Done.

Mode 4 (Address register indirect with pre-decrement)

Add '-' to the op-code buffer.
Call Mode 2 processing to add '(An)' to the buffer.
Done.

Mode 5 (Address register indirect with displacement)

Add a '$' to the buffer - hex data coming next.
Convert the word at (A6)+ to hex and add to the op-code buffer.
Call Mode 2 processing to add '(An)' to the buffer.
Done.

Mode 6 (Address register indirect with index)

Add a '$' to the buffer - hex data coming next.
Get the word at (A6)+ into D0 - index and displacement details.
Copy D0.W to D4.W.
Convert D4.B to hex and add to the op-code buffer.
Add '(A' to the op-code buffer.
Convert D1.W to hex and add to the buffer.
Add ',' to the op-code buffer.
Copy D0.W to D4.W again.
Call index register routine.
Done.

Mode 7 (Others).

Mode 7 is not a 'valid' mode. It flags that a sub-mode is in the register part of the effective address. We now need another jump table to get us to the correct routine to process these sub modes. At present, the registers look like the following:
D0.W = mode * 2
D1.W = Register number = sub-mode number
D2.W = Don't care
D3.W = Don't care
D4.W = Don't care
D5.W = Size of instruction (1, 2, 4 for byte, word, long)

D6.W = Length of text in op-code buffer (so far)
D7.W = op-code word.

A0.L = Don't care
A1.L = Don't care
A2.L = Don't care
A3.L = Don't care
A4.L = Our programs dataspace pointer
A5.L = op-code buffer pointer
A6.L = Next word in memory after op-code word
A7.L = Stack pointer

Point A3 at sub_modes table
Shift D1.W one bit left. (Double it in other words)
Get the word at A3.L + D1.W into D2.W.
Add D2.W to A3.L to get the address of the mode sub-routine.
Jump to the appropriate mode routine. (Return to here so use JSR not JMP)

```
sub_modes    dc.w    sub_mode_0-sub_modes
             dc.w    sub_mode_1-sub_modes
             dc.w    sub_mode_2-sub_modes
             dc.w    sub_mode_3-sub_modes
             dc.w    sub_mode_4-sub_modes
             dc.w    sub_mode_5-sub_modes
             dc.w    sub_mode_6-sub_modes
             dc.w    sub_mode_7-sub_modes
```

Although only the sub-modes 0 to 4 are defined by Motorola, we must check for illegal values in D1 which can have a value between 0 and 7. Modes 5, 6 and 7 simply output an error message.

There is no need to preserve D0 or D1 but the rest must be correctly updated or preserved especially D5.

Sub mode 0 (Absolute short address)
Add a '$' to the buffer - hex address coming next.
Convert the word at (A6)+ to hex and add to the op-code buffer.
Done.

Sub mode 1 (Absolute long address)
Add a '$' to the buffer - hex address coming next.
Convert the long word at (A6)+ to hex and add to the op-code buffer.
Done.

Sub mode 2 (Program counter with displacement)
Add 'L' to the op-code buffer.
Copy A6.L to A3.L
Add the word at (A6)+ to A3.L (sign extended)
Convert A3.L to hex and add to the op-code buffer.
Add '(PC)' to the op-code buffer.
Done.

Sub mode 3 (Program counter with index)
Add 'L' to the op-code buffer.
Copy A6.L to A3.L
Get the word at (A6)+ into D0 - index and displacement details.
Copy D0.W to D4.W
Add D4.B to A3.L (sign extended)
Convert A3.L to hex and add to the op-code buffer.
Add '(PC,' to the op-code buffer.
Copy D0.W to D4.W again.
Call index register routine.
Done.

Sub mode 4 (Immediate data)
Must preserve D5.W and D0.W as called from Type 18 (etc) above.

```
Add '#$' to the op-code buffer.
If D5.B = 4 (Long)
   convert the long word at (A6)+ to hex and
   add to the op-code buffer
else
   get the word at (A6)+ into D4.W
   if D5.B = 1 (byte)
      convert the byte in D4 to hex and add to the op-code buffer
   else
      convert the word in D4 to hex and add to
      the op-code buffer
   end if
end if
Restore D5.W and D0.W
Done.
```

Sub modes 5, 6 and 7 (Illegal sub-modes!)
These modes cause an error message to be logged in the output text.

Add 'M7,R' to the op-code buffer.
Convert the byte in D1 to hex and add to the op-code buffer.
Add ' <--' to the op-code buffer.
Add 'DATA' to the op-code buffer.
Add ' ???' to the op-code buffer.
Done.

Index register routine

INPUT:
D4.W = D0.W = index/displacement word.

OUTPUT:
Used for d(An,Xn) and d(PC,Xn) instructions.
Adds the bits after the comma to the buffer.

```
If bit 15 of D0 is set
    Add 'A' to the op-code buffer.
else
    Add 'D' to the op-code buffer.
end if
Mask out all but bits 12 to 14 of D4.
Shift D4.W right 12 bits.
Convert D4.B to hex (0 to 7) and add to
op-code buffer.
If bit 11 of D0 is set
    add '.L)' to the op-code buffer
else
    add '.W)' to the op-code buffer
end if
Done.
```

Finally ...

So that is it for part 4. In Part 5 we shall be updating the main disassembly loop to decode which type of instruction is being decoded, set up all the registers and call the above routines. We will have to convert the above pseudo-code into assembly language as well. More typing!

For homework, why not try converting a few routines by yourself and see how they match up when we come back again.

See you then.

Test release of TCP/IP for the QL at QL2000

Jonathan Dent

This article doesn't look much different from the other articles in QL Today this month or in previous months. It is however different as it was sent in to Jochen by email using soqlMailer. This is a version of Jonathan Hudsons qlmailer which he wrote originally for UQLX. On UQLX it uses the underlying TCP/IP stack of UNIX to access the Internet. SoqlMailer uses the TCP/IP stack of the soql Internet access software. This means it is now possible to send and receive emails with any QL based system which has a reliable serial port.

Application programs like soqlMailer access the Internet by using functions contained in the SOQLlib_a library for C68. These functions allow the application programs to communicate with the soqlEngine, a program which runs at the same time as the applications and feeds data from the Internet to application programs and vice versa.

To send emails soqlMailer has to get in touch with an SMTP mail server somewhere in the Internet. SMTP stands for simple mail transfer protocol and the server and soqlMailer use this protocol to enable emails to be transfered to the SMTP host from the QL.

When you've sent someone an email he may well send a reply. This should arrive at your POP3 server. To fetch the mails from there you use a POP3 client program. POP3 stands for post office protocol 3 and there is an experimental program pop3 included in the test release.

So, what do you need in order to exchange emails in Internet

Hardware

I have experimented with different QL derived systems. The main problem with the original QL is the serial ports. Reliable communication is only possible up to 1200 Baud. Above that rate the data flow has to be regulated by the use of the DTR signal (ser2) for example. This is fine for computer to computer connections just connect CTS on the other computer to the DTR signal and the flow can be controlled on a byte by byte basis. Unfortunately that won't work with a modem, you can't control the data flow from a modem. This means you need to up-grade your serial ports. DTR is normally used to hang-up the telephone connection. The modem at my ISP won't connect at speeds below 9600Bd so an unmodified QL just ain't good enough.

With SuperHermes from TF-Services Ser2 will work reliably up to 9600Bd. For even higher speeds SuperHermes Ser3 will have to be pressed into service. This requires some extra wiring.

My SuperHermes is in my SuperGold card computer together with Aurora. Bruce Nichols has reported that the soql-System works on Qemulator and QPC. How it works on other platforms has yet to be seen.

The final hardware requirement is a modem. I use an E-Tech Bullet. Bruce used a US-Robotics but there is no reason why any modem for rates between 57600Bd and 9600Bd should not work.

Software

The complete software fits easily on a DD disk. The software runs on Minerva and Sinclair ROM systems and on

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://www.j-m-s.com/smsq/index.htm

QPC2 Version 2 Final is available!

We deliver since end of December!

A special goody has been added to say "Thank you for your patience" - the DOS device. You can now read and write from and to DOS and Windows harddisks. DOS1_ = C:, DOS2 = D: and so on!

We have told you about the new features many times in advance: they are all fully functional: QPC2 is now running inside a Windows window (but, of course, full screen can be used as well). You have access to 65536 colours etc.

To avoid confusion: If you already own QPC2 (Version 1), then the upgrade is NOT free!. The naming is a bit confusing (we should have called it QPC3, but now it is too late, isn't it?), so here is the complete price list:

Upgrade to QPC2 Version 2 Final

From QPC1 (or just QPC - the first QPC!)	£41.90
From QPC2 Version 1 (or just QPC2)	£25.90
From QPC2 Version 2 any Pre-Release	FREE!

You can get free Updates in the JMS BBS (+49 203 502013 and 502014)
You will also find the free Update on the Internet: http://qpc.j-m-s.com

TERMS OF PAYMENT

Postage and package **[Germany]** DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99).
[Europe] £4,50 (if total value of goods is under £15 then only £3).
[Overseas] between US$7.50 (1 item) and US$17.50 (maximum).
All prices incl. 16% V.A.T. (can be deducted for orders from non-EU-countries).
Cheques in DM, EURO, Eurocheques and Credit Cards accepted.
Please note: Prices are based on an average exchange rate of £1 ranging between DM 3,10 and DM 3,20. Prices may be adjusted in case the exchange rate falls out of this range - in both directions!
We can charge your credit cards in £'s, US$, EURO or in DM - please state the currency you prefer.
Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque. E&OE.

EUROCARD VISA MasterCard

SMSQ-E and should run on other QL based operating systems.

Internet Portal

The present version requires an ISP (Internet Service Provider) who supports SLIP (Serial Line Internet Protocol) portals. In England Demon is the best known ISP who does this. SLIP is the absolute simplest protocol for serial lines connecting to the Internet. Many ISP's (especially free ones) now support only PPP or Point to Point Protocol. If PPP is a must for you then you'll have to be patient somewhat longer. PPP for soql is still in development. If you'd like to be counted among the pioneers of QL email then get yourself a SLIP ISP and although you may well have to pay for that you can have a (pre-)Beta copy of the program for free. I distributed about 10 disks at QL2000 in Portsmouth and now have a new version which uses an improved SOQLlib. This will be the basis of a standard library which Jonathan Hudson is preparing. The aim of this standard library is to promote the porting of programs from other platforms.

Want to give it a go?

Check-out your local ISP's; can you drum up a SLIP portal? Talk to someone technical if you can, rather than the sales staff. I'm sure the readers of QL-Today will be pleased to hear of any successes. If you send me international reply paid coupons and your address I'll send you a disk or I can send it via email. Please then send an email to **jondent@crosswinds.net** to let me know how you get on.

If your interested but would prefer to wait for the PPP version then don't hesitate to get in touch so that I can gauge the level of interest.

Jon. Dent
Brueoelring 1A
6415 Arth am See
Switzerland
Email: **jondent@crosswinds.net**

Programming ProWesS in Sbasic - Part 5

Wolfgang Lenerz

As promised last time, in this instalment we'll have a closer look at loose items. Loose items are small "buttons" on which you can click with the mouse, either hitting or doing the item. You can also indicate the loose item with the keypress that corresponds to it. Both of these actions will usually result in some kind of action. Thus, clicking the "QUIT" loose item in a window will generally result in the program's quitting.

Why two types of loose items?

There are actually two types of loose items provided in Prowess, even though only one can be used at the same time. These are the 'item' type on the one hand and the 'itemp' type on the other hand. Both types of loose items are very similar, except in one point: When the pointer appears over an "item" type loose item, a border is drawn around it. This is used to show that this item is the one that currently has 'focus', which means that clicking the mouse will have an action on this item, and not on anything else.

The behaviour is different when using the "itemp" type item: No border is drawn around it even when it has focus.

The choice between both is left to the user (NOT the programmer!!!). How can you choose between these types? Simple, in the Prowess_cfg file, which MUST be present in your system. It is usually located in the _mine_ subdirectory of your prowess directory. If you have a look at this file (with a texteditor, it is a simple text file) you can see lines starting with a 'T' - these indicated to Prowess which Types it should load. You have two lines, one saying "T item" and the other "T itemp". One of these is commented out with a semicolon (usually T itemp). Depending on which one you leave in, you have either one or the other type.

More on loose items

Loose items normally contain some text, although a routine can be provided to draw an icon or a sprite in them.

A loose item usually can react to both HIT and DO events. If you want, you can also attach a keypress to it, which will be the same as a HIT event.

Statusses

A loose item can have one of three statusses: available, selected or unavailable. When an item is ***unavailable***, it can't be indicated (and thus will not get a border). When the item is ***available***, you can Hit or Do it. A HIT on the item will usually toggle between available and ***selected***, and a DO will select the item.

However, the programmer can also determine that the status of a loose item can never change.

Action routines

A user-defined function can also be called on a HIT or DO event (but not when the item is unavailable). Under S-Basic, these routines are actually very important, because only they can cause a return to the program. You will remember that all of the work within Prowess is done when you **PWactivate** the window. Some events (such as the user clicking on the Quit item to suit the program) cause a return from the **PWactivate** function.

However, we also want other events to cause a return from this function - such as a user clicking on an item, because then we must execute the action corresponding to the item, and to do this, we must return to S-Basic. The only way you can generate such a return is to indicate an "action routine" for the items. If you don't, no return will be made to S-Basic. Note, in some cases, it is not necessary to return to S-Basic when an item is hit or done, if the only purpose of the item is to show some state (i.e. the item is selected or not).

Normally, Prowess expects that the "action routine" that is passed to it for an item is the routine that will really be executed when the item is indicated. In Basic, this would mean that we pass it the procedure or function that is to be called when the item is hit/done, and that, of course, is impossible. Instead, the S-Basic interface provides some new keywords, the "action routines" keywords. These can be passed instead of the routines which Prowess expects, and they will cause a "fake "return to S-Basic.

These action routines are thus used whenever a tag allows an action routine be passed as a paramater. This is the case, for example, for the OUTLINE_ACTION_DO, OUTLINE_ACTION_INFO etc... tags, which we saw last time. For all of these tags, you pass a parameter which is a "routine", which returns to S-Basic and tells you what routine was called.

In SBasic, you pass HIT_ROUTINE, DO_ROUTINE or any of the other xxxxx_ROUTINEs as parameters. These are actually functions which return an address (try: "Print HIT_ROUTINE") and it is that address that is passsed on to ProWesS. At that address lies a routine that does the fake return to SBasic. When HIT_ROUTINE comes back, it returns a certain value in the hit_or_do% parameter to the **PWactivate** keyword, whereas DO_ROUTINE returns another value. EXIT_ROUTINE returns yet another value and so on... This allows you to check which routine was called for a certain object by SELecting on the hit_or_do% parameter which is changed by the **PWactivate** function.

The different action routines

Here is a list of the different routines and the values they return. Please note that all routines are not possible for all of the objects: a ***loose_item*** object, for example, has no provision for you to set a redraw routine...:

HIT_ROUTINE
This should be used whenever an action routine for a HIT on an item is required. The value returned (in the hit_or_do% parameter) is 0.

DO_ROUTINE
This should be used whenever an action routine for a DO on an item is required. The value returned (in the hit_or_do% parameter) is 1.

RDRW_ROUTINE
This should be used whenever an action routine redrawing an item is required. The value returned (in the hit_or_do% parameter) is 2.

EXIT_ROUTINE
This should be used whenever an action routine is required for the event that the pointer leaves an object. The value returned (in the hit_or_do% parameter) is 3.

MOVE_ROUTINE
This should be used whenever an action routine is required when a MOVE_EVENT is captured by an object. The value returned (in the hit_or_do% parameter) is 4.

SCALE_ROUTINE
This should be used whenever an action routine is required when a SCALE_EVENT is captured by an object. The value returned (in the hit_or_do% parameter) is 5.

DRAGH_ROUTINE
This should be used whenever an action routine is required when the user starts dragging with a Hit. The value returned (in the hit_or_do% parameter) is 6.

DRAGD_ROUTINE
This should be used whenever an action routine is required when the user starts dragging with a Do. The value returned (in the hit_or_do% parameter) is 7.

DRAGE_ROUTINE
This should be used whenever an action routine is required when the user Ends dragging. The value returned (in the hit_or_do% parameter) is 8.

DRAGA_ROUTINE
This should be used whenever an action routine is required to trap the pointer moving out of the object, and making an Adjustment. The value (in the hit_or_do% parameter) returned is 9.

INFO_ROUTINE
This should be used whenever an action routine is required for an info item. The value returned (in the hit_or_do% parameter) is 10.

WAKE_ROUTINE
This should be used whenever an action routine is required for a wake item. The value returned (in the hit_or_do% parameter) is 11.

QUIT_ROUTINE
This should be used whenever an action routine is required for a quit item. The value returned (in the hit_or_do% parameter) is 12.

HELP_ROUTINE
This should be used whenever an action routine is required for a help item. The value returned (in the hit_or_do% parameter) is 13.

INIT_ROUTINE
This should be used whenever an action routine is required following a PW('SYSTEM_ACTION_INIT') tag. The value returned (in the hit_or_do% parameter) is 14.

EVENT_ROUTINES
There are 8 event routines - EVT1_ROUTINE to EVT8_ROUTINE. They can be used for the system events. The values returned are 21 to 28.

There is, of course, nothing to stop you from using a HIT_ROUTINE for a DO_ROUTINE and visa-versa, if you wish. All of these routines are essentially the same, they only differ in the value they return. So, provided you adjust your SELect, you can use any routine for anything you want. I would suggest, however, to use the routines for their purpose, to avoid confusion...

Example:

```
(...)
object=PWcreate (0,(...),
PW('OUTLINE_ACTION_WAKE'),WAKE_ROUTINE,
PW('OUTLINE_ACTION_INFO'),INFO_ROUTINE,
PW('OUTLINE_ACTION_DO'),DO_ROUTINE,
(...)
```

Here, indicating the wake item will call the WAKE_ROUTINE which returns to SBasic, filling in the hit_or_do% parameter with 11. The **object_hit** will be the Wake object (you must previously have queried the Outline object to know what the object identifier of the wake object is, so that you can SELect on it).

Back to the loose items

Ok, so now we know what the different action routines are and do. Now we only have to investigate the different tags for a loose item, to see what is possible with this Prowess type. As usual, change tags can also be used when creating the object, but query tags are only used for queries.

The change (and creation) tags

PW('LOOSE_STATUS')
Set the current status of the loose item. This tags needs one parameter, which can be either *PW*('STATUS_AVAILABLE'),*PW*('STATUS_UNAVAILABLE') or *PW*('STATUS_SELECTED'). If the new status if different from the old one, before attempting to set the status), then the item will be redrawn when control is handed back to ProWesS. By default, the item status is set to *PW*('STATUS_AVAILABLE').

PW('LOOSE_TEXT')
Set the text which should be displayed inside the loose item. The parameter is a string. When AUTOSIZE is true, then the size of the loose item will be redetermined when control is handed back to ProWesS. In other words, a check is made to see whether the new text will fit in the item as is. If not, the item will be made bigger so that the texts fits in it. In that case, the whole window will be redrawn since the size of the item may have changed.

PW('LOOSE_TEXT_COPY')
Set the text which should be displayed inside the loose item. The parameter is a string. The text is copied into a piece of memory which is allocated (and released) by the loose items itself, so this can be a direct string. When AUTOSIZE is true, then the size of the loose item will be redetermined when control is handed back to ProWesS (see above).

PW('LOOSE_CHANGE_STATUS')
The parameter allows you to determine whether an item can change status or not. The parameter necessary for this tag is either 1(=TRUE) or 0 (=FALSE). By default, the value is TRUE. The status of the loose item is only changed when change status is TRUE. In this case a HIT will switch between *PW*('STATUS_AVAILABLE') and *PW*('STATUS_SELECTED'). In the case of a DO, the status will always be set to *PW*('STATUS_SELECTED'). When the loose item is unavailable (its status is set to *PW*('STATUS_UNAVAILABLE')), the status is not changed automatically (actually, not even a border will be drawn around the item).

PW('LOOSE_WINDOW_DO')
The parameter is either 1 (=TRUE) or 0 (=FALSE). By default it is FALSE. When the window do status is TRUE, then the ENTER keypress is also handled by the system (thus a keypress object can react to it).

PW('LOOSE_CENTER_ITEM')
The parameter is either TRUE or FALSE. By default, the value is TRUE. If the value is TRUE, then the text for the item will be drawn in the centre of the loose item, else it is drawn in the top left corner.

PW('LOOSE_AUTOSIZE')
The parameter is either 1 (=TRUE) or 0 (=FALSE). By default, the value is TRUE. When autosize is TRUE, then the size of the loose item will automatically be redetermined when the text inside the item is changed. In other words, Prowess makes sure that the texts fits in the item. If autosize is FALSE and no size is set explicitly, then the size of the text in the time when the window is first activated is used.

PW('LOOSE_ACTION_HIT')
Set the routine which should be called when the loose item reacts to a HIT. The parameter should be an action routine, preferrably HIT_ROUTINE.

PW('LOOSE_ACTION_DO')
Set the routine which should be called when the loose item reacts to a DO. The parameter should be an action routine, preferrably DO_ROUTINE . Please note that, if no DO action exists for the loose item, then the HIT action routine will be called (if it exists) instead, even on a DO!

PW('LOOSE_ACTION_DRAW')
Set a draw action for the loose item, the parameter should be an action routine, preferrably ***RDRW_ROUTINE*** or ***PWsprite***. With this, control will be passed back to SBasic whenever the item is to be drawn on the screen (e.g. when the window is opened). This means that you now can use, for example, Proforma to draw some kind of icon or sprite in the item. The (Proforma) Subwindow is set to cover the hit area of the loose item before control is passed to the action routine. If the item also contained a text, this will already be drawn. This makes it possible (in combination with the tags to set the size) to draw icons in loose items. Please see below, under the PW('LOOSE_SPRITE') tag, in what order the content of an item is drawn.

PW('LOOSE_SIZE')
Set the minimum width and height of the loose item. The tag needs two parameters, the x and y

If our software is friendly to you...

"Intuitive and clear"

"A very useful labour saver"

"On-screen help is concise"

"Navigating is point and click"

"Actually kind of fun"

"Manual is simple to follow"

"Will save you valuable time"

"Almost instantly returning results"

...you are friendly to us

Geoff Wicks, 28 Ravensdale, Basildon, Essex, SS16 5HU, UK.
Tel: +44 (0)1268 281 826   Email: geoffwicks@hotmail.com
Web: http://members.tripod.co.uk/geoffwicks/justwords.htm

Just Words! - Software for Writers and Word Lovers.

size, both in PROforma numbers. Setting the size automatically also sets AUTOSIZE to FALSE. This seems logical: normally, you leave it to Prowess to determine what size objects have. This is why Autosize exists, to make sure that everything fits nicely together in the window. If you circumvernt that, by requiring a specific size for the item object, Prowess now longer cares about it so much, and sets Autosize to False.

PW('LOOSE_XSIZE')
Set the minimum width of the loose item. The tag needs a PROforma number as parameter. Setting the size automatically also sets AUTOSIZE to FALSE.

PW('LOOSE_YSIZE')
Set the minimum height of the loose item. The tag needs a PROforma number as parameter. Setting the size automatically also sets AUTOSIZE to FALSE.

PW('LOOSE_KEYPRESS')
Attach a keypress to the loose_item. Hitting that key will be equivalent to a HIT on the item. The parameter is the primary keypress, which is of type CODE (character$).

PW('LOOSE_AUTOREPEAT')
When this tag is passed, then HITting and DOing on the item will autorepeat, i.e. keeping the mouse buttons, or ENTER & SPACE pressed will repeatedly perform the action for this item.

PW('LOOSE_SPRITE')
This sets a sprite for this item. The parameter is a number, which is the address of the sprite. The sprite itself is a normal, mode 4 Pointer Environment sprite, which is preceded, however, by a 60 byte header. The header is mostly empty, except that the first bytes should contain the name of the sprite, which should end in _sp4.
Once this is set, the sprite will be drawn whenever the item is to be drawn. The sprite will always be centred in the item, and will be made as big as possible within the item, provided that it fits correctly in both dimensions.
There are thus several ways to set the content of an item: you can set a text, you can set a sprite, you can use another drawing routine which is set with **PW**('LOOSE_ACTION_DRAW') - or you can use any combination of this! Indeed, please note that drawing order is as follows:

- First, if it exists, the text in a sprite will be printed.
- Then the sprite will be drawn, if it exists.
- Finally, the external drawing routine will be called, if it exists.

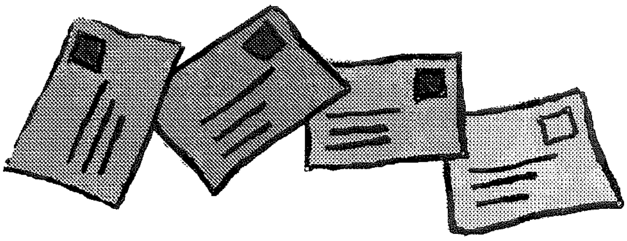
The query tags

Here are the query tags for loose items - there are only a few!

PW('LOOSE_STATUS')
Get the current status of the loose item. The status can be either **PW**('STATUS_AVAILABLE'), **PW**('STATUS_UNAVAILABLE') or **PW**('STATUS_SELECTED').

PW('LOOSE_TEXT')
Get a pointer to the text which is displayed inside the loose item. This text is read only! Use MKSTRING$ to make that pointer into a normal SBasic string. Normally, you wouldn't need this, if you keep the text in a variable anyway.

Letter-Box



P.Selhorst writes:
In recent issues of QL Today some people declared their love of QPC and I fully understand that. I used my 3 QL's every day for almost 15 years and during that time I became very dependent on them but it was a continuous battle to keep the system afloat.
At this moment I have a big box filled with all kinds of obsolete and defective hardware. I could organise a complete QL museum on my own. Almost everything that appeared on the QL scene is in that box: microdrives (8), harddisks (5), ED drives (4), keyboards (3), eproms (...), interfaces (...), etc.
The QL network was of great value to me especially for backing up large files of more than 1.7 Mbytes. The QL NET caused me a lot of problems e.g I had to 'CACHE_OFF' all computers for reliable operation.
I year ago I bought 3 PC's and linked them together with a fast 3COM network. I installed QPC2 (v1.54) and tested SERNET but this only worked on rare occasions so I had a problem. It was never clear to me what caused the serial problems but tests with simple SBASIC programs reveilled unreliable serial ports.
My first solution was to copy the complete

QXL.win file from PC1 to PC2 or PC3 or vica versa by using EXPLORER. Tranferring 30 Mbytes takes only 5 minutes!

Since a week ago there is a second (magic) solution: I configured qpc2 on PC1 and PC3: win1_ as c:\qxl.win and win3_ as \\pc2\c\qxl.win and removed QPC2 (QPC_EXIT) on PC2. File tranfer from one QXL.WIN file (PC1 or PC3) to the other QXL.win (P2) is now possible. All file operations are possible: wcopy, dir, ex, bput, lrespr etc. Tranferring 1.7 Mbytes takes only 15 seconds!

It is even possible to share files between computers at the same time but QPC2 on PC2 (the one that owns the shared QXL.win file on its harddisk) must be non-active.

To resume: PC1 and PC3 (both QPC2 active) use QXL.WIN on PC2 (no QPC2).

Unfortunately modems, printers, ramdisks, consoles etc cannot be used in this way.

J.C. Marcus writes:

Firstly, I would like to thank you and all those who make QLT possible. Users like myself would be completely lost without all the hours and effort you and all those still commited to the QL/SMSQ machines put in on our behalf.

Ok - so much for the grovelling!

In the November/December 2000 QLT, Geoff Wicks has an article about hard drives, which has (more or less) prompted me to write this. Now, despite all their many advantages, hard disks seem to have two large disadvantages:

1. they will inevitably crash, losing M-bytes of data!
2. they are a pain to back-up, and one tends to get lost in mountains of floppies while doing this, unless you fork out for a tape-streamer system or similar.

Well, I think that I have found a simple but effective solution to these problems for those with a similar set-up to my own.

To begin at the beginning, I bought a Qubide interface and hard disk from Ron Dunnett a few years back, although I never actually got around to fitting the drive until the Aurora came out, and everything (Aurora, SGC, Qubide & HD) went into a tower case. I then spent several joyful hours making my sub-directories and passing everything from floppies onto the HD. This blissful state lasted about a week, then the HD started going wrong, finally crashing and sinking without trace.

After I'd recovered from the shock, removed the drive from the case, and jumped up and down on it for a few minutes, I took a step back to think what I should do next. At the time, the 2nd-hand market here was pretty much non-existent and all the shops here offered were the latest 2.3Gb HD's for the PC. The thought of connecting anything that large to a QL seemed ridiculous to me, and although a friend in London offered to find another 2nd-hand HD for me, my previous experience had made me somewhat cautious over buying something without any guarantee.

I then remembered reading something about the Iomega 100Mb Zip drives. A quick phone call to Ron assured me that it "should work", so I splashed out. For those not familiar with the drive, it comes in two versions. The "internal", which is Qubide compatible and fits into a standard 3.5" floppy bay, and the "external", which is a stand alone unit which I never really bothered looking at, but I think needs a bi-directional parallel port, but don't quote me on that.

[The original external 100MB zip drives were available with parallel or SCSI interfaces, more recent ones are also available with USB or PC Card interfaces - editor]

I now have one 100Mb Zip disk as my main "boot--up" disk, another as the backup and one more for my most important data, and other odds and ends. Simple, but effective. Also, so far, the drive has been 100% reliable and has never given me any problem whatsoever (and I don't have anything to do with the company, honest).

Ian Pizer writes: GD2? About Colours??

GD2 refers to the new Graphic Device Interface Version 2 which you can read about by downloading Tony Tebby's GD2doc.zip from Thierry Godefroy's WEB site.

Unzipped you find 3 .DOC files and 3 .HTML files. The .DOC files are in Microsoft WORD format but you can find ways to read them in your "QL".

[You could try converting them with Catdoc from Jonathan Hudson's website - editor]

However, to benefit from colours you need QPCv2v2f and SMSQEv2.98 so you must use a PC anyway unless you have a Q40!

With a PC you can read the HTML files by using your Web Browser in offline mode and opening the HTML file into the browser.

In GD2-USERS.doc you learn how to change the background (BGCOLOUR_QL, BGCOLOUR__24) and choose colours for ink (COLOUR_QL, COLOUR_PAL, COLOUR_24) You will need to study the documents to be fully informed.

To be impressed by colours run the following BASIC program:

```
100 CLS:y=2:FOR x=255 to 500: BLOCK#1,500,20,0,
y,x: AT 20,30: PRINT#1,x: PAUSE 50: y=y+2
```

From there you can begin to be inventive with the possibilities. I do not yet have a colour printer so cannot check that printing colours works from BASIC without a suitable colour driver.

Prowess provides a colour driver for EPSON printers so printing in colour from Paragraph should work and also from Line Design if eventually colour is provided for text and images.

Have fun with colours!

SOUND DEVICES for Minerva, SMS-Q40 and QDOS CLASSIC

Simon N. Goodwin

This article explains how the new SOUND device for QL-compatibles works, and discusses some of the utilities you can use to capture and play sound on QL, SMS-Q40, UQLX and Qdos Classic systems. Those who attended QL2000 last year should be aware that a variety of high-quality samples are now available for this system - I have sent copies of them to England, Scotland and Wales (Roy Wood, George Gwilt and Dilwyn Jones respectively) and I hope they will soon appear on one of the Qdos CDs, as they total about five megabytes of data.

I seized a gap late last year between real jobs (and nappy changes) to write a device to allow samples to be PRINTed or COPYed to the sound hardware on the Amiga, Q40, and any other Qdos compatible system that supports sampled sound output. This freely-distributable work was prompted by Mark J Swift's implementation of the Sampled Sound System in Qdos Classic, following work on SMS for the Q40, and the UQLX emulator for Linux. The person who wrote that, Jonathan Hudson, also wrote a task to drive it, and made reference to a 'relatively trivial device driver'. This spurred me to action, along with a promise to write a new device to show Tim Swenson why it's sadly not something you can do in SuperBASIC, although otherwise not that complicated.

The source code for the SOUND device is about 16K of Devpac assembler, and the device driver code is just 680 bytes long. This includes nine different routines output routines, for mono and stereo replay at various sample rates - the actual output to the hardware is done by interrupt service routines in QLSSS, the QL Sampled Sound System, which is part of Qdos Classic and SMS-Q for the Q40. These have a standard interface so new hardware that supports the same three machine-code routines should work just as well with my device driver. The SOUND device is easy to use from SuperBASIC PRINT and COPY commands, but potentially compatible with all programs capable of writing to a standard Qdos device.

I know Goldfire designer Zeljko Nastasic has a lot of digital audio experience and I hope he will add to the range of Qdos-compatible systems that can play samples. I've been writing sound utilities and applications for Spectrum and Amiga computers for most of my career, and I'm currently in charge of the team developing a custom audio solution for the next generation of Amigas, but this is an area where the QL has lagged behind other systems. The first QL sample-playing programs I wrote were for the I2C interfaces developed by Tony Firshman; I once heard a demo of a DiRen QL sound card but have never had the chance to program it, and gather it is not currently available.

Minerva samples

The Analogue Interface for the Minerva add-on is capable of playing eight bit samples, although the serial interface

Sound Theory

Sampled sound uses a sequence of values output at regular intervals to describe a wave. The values are converted to voltages by a DAC (Digital to Analogue Converter) and corresponding sound waves are generated when that signal is amplified and fed to headphones or a loudspeaker. The limit on quality is set by the number of samples per second, and the resolution of each sample. The QL beeper has only one bit resolution - on or off - so all the sounds it can generate are square waves. The 8049 co-processor can vary the timing of these, but not their aplitude, which is why it can generate raspy tones but has no volume control (other than the externally-applied sock recommended by SuperBASIC author Jan Jones) and even with new software it would be useless for speech or pre-recorded music.

Telephone circuits use seven or eight bits, giving one or two hundred levels at a rate of 8000 samples per second. This is good enough for clear speech and recognisable music, though the low rate means that the top couple of octaves of notes a human can hear cannot be reproduced on a telephone - the effect is most noticable on cymbals but makes most sounds duller than they would be if the sample rate was higher. Telephone companies deliberately limit the rate so they can get more calls down a trunk line, and charge extra for faster digital connections like ISDN and ADSL, which use more signal 'bandwidth'.

Analogue telephone lines use only 7 to 8K per second of bandwidth, and that determines the upper limit for modem speed; 7K is 57,344 bits, and if a modem manages to put 56,000 of these to good use you're doing exceptionally well!

CD players deliver much higher fidelity because they have more than two dozen times as much data to work with - they deliver 44,100 pairs of samples, each accurate to 16 bits (65,536 levels) every second.

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost.. £24

**************NEW***************

superHermes

A major hardware upgrade for the QL

All Hermes features (working serl/2 at 19200, independent baud rates/de-bounced keyboard/ keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)	**£90** (£92/£93)
IBM AT UK layout Keyboard	**£11** (£13/£15)
Serial mouse	**£8** (£8.50/£9)
Capslock/scrollock LED	**£1** (£1.50/£1.50)
Keyboard or mouse lead	**£3** (£3.50/£3.50)
High speed serial (ser3) lead	**£4** (£4.50/£4.50)

Hermes available for £25 (£26/£27) **Working serl/2 and independent input, debounced keyboard.**

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead).....................**£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI...£40 (£41/£43) **MKII...£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.
Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq	**£39** (£40/£41)
4mbytes RomDisq	**£65**(£66/£67)
8 mbytes RomDisq	**£98** (£99/£100)
Aurora adaptor	**£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/ Goldfire fixed to base. Suitable for Aurora (ROM acccssible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost .... £34 (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**

2 amp (for 8 relays, small motors) .... **£40** (£43/£44)

4 amp total (for motors etc) .... **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..**£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.........**£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting.................**£30** (£31/£32)

Temp probe (-40°C to +125°C)..............**£10** (£10.50/£11)

Connector for four temp probes.............**£10** (£10.50/£11)

Data sheets...**£2** (£2.50/£3)

Control software & manual (for all I/F).......**£2** (£2.50/£3)

QL SPARES

Keyboard membrane	**£24** (£25/£26)
1377 PAL	**£3** (£3.50/£4)
Circuit diagrams	**£3** (£3.50/£4)
68008 cpu or 8049 IPC	**£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead	**£10** (£10.50/£11)
Power supply (sea mail overseas)	**£12** (£17/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address,/ postal order, Giro transfer (58 267 3909) or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list 13 Nov 00

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

limited it to quite a low sample rate - the upper limit is 10,000 samples per second in mono, or half that in stereo, giving audio quality somewhere between that of telephone and AM radio.

The I2C analogue interface is supplied with a function, WRITE_adc%, which allows SuperBASIC programs to spew a stream of samples to the hardware. The 32K limit on the length of strings means no more than about three seconds of audio can be output in one go, though you can send a sequence of samples in rapid succession.

My contribution was a handful of SuperBASIC programs to make it easier to manipulate samples. I wrote a converter to read samples in a couple of standard formats - Amiga IFF 8SVC (Interface File Format 8-bit Sampled VoiCe) and Microsoft WAV(e) format - and convert them to eight bit uncompressed unsigned values. Since I did this Jonathan Hudson has ported a heavyweight Unix program, SOX, to Qdos - SOX supports more formats and sample rates, but takes quite a while to load and run on a typical QL, as the package is half a megabyte in size, when ZIPPed.

My little programs convert samples on a one-to-one basis, so audio recorded at another rate will play fast or slow on the QL, and data that has been compressed or recorded in some variant of the usual format will not be decoded. However they work well with the majority of the thousands of small freely-available samples you'll find in disk libraries, on CDs and the net, and if you want to extend them you have the SuperBASIC source to edit as you wish. The conversion routine is only about a page long, in each case, including comments and error traps.

I've also written a trivial routine to plot the graph of a wave in sections across the QL screen. This can be handy when you want to edit the beginning or end of the wave to save space, as you can easily see the loud parts. Sample levels can be faded up and down by multiplying each value by an appropriate factor, and you can even mix samples together by adding values. If the total is likely to exceed 255 you'll need to divide the output by a factor that ensures it remains in the eight bit range 0 to 255. These programs, and some sample sounds, have been donated to TF Services and thus are available from Tony Firshman.

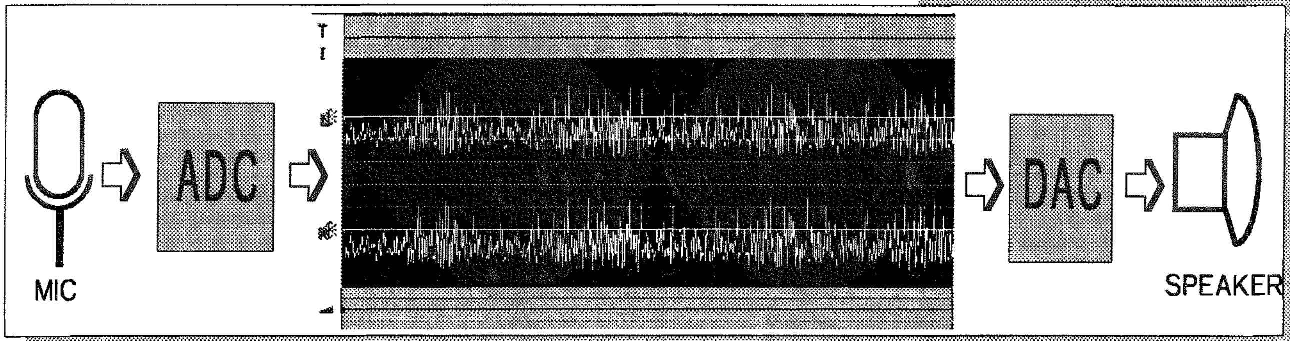
NEW SOUNDings

It's been several years since I worked with I2C audio, though the hardware remains available from TF Services, and can do a lot more than replay samples. In theory my new SOUND device could be adapted for I2C, but the low sample rate would limit its usefulness, so I've opted to make the new device

The snag of this is that the data rate is 172K second, so you need ten megabytes of disk space for each minute of audio at CD quality.

Existing QL sampling systems work at intermediate rates and resolutions. There is a trade-off between sound quality, number of channels against the need for fast hardware and storage space. I2C can only deliver 10K per second, wheras Q40 hardware can write 20,000 samples per second (20 KHz) to each channel - twice the I2C rate in mono, and four times that in stereo.

The Amiga hardware supports a wider range of sample rates, continuously variable from a few Hertz (samples per second) to higher than CD rates, but its output of samples is synchronised to the video beam (at eight values per scan line, in four channels) so the upper limit is about 28 KHz on a TV or compatible monitor like a Philips CM8833; Amiga Qdos emulators do not yet support faster multisync displays, so the upper limit is restricted. However you can tweak the rate up and down by poking values to the period registers at 14676134 and 14676150 - the default for the standard QLSSS rate is 177 for PAL displays, and 177 for the NTSC used in Japan and the USA. If you reduce these to 161 or 162 respectively with POKE_W you can boost output to half the CD rate, around 22 KHz, and play samples captured from CD at their proper pitch by using devices SOUND6 to SOUND9. On a fixed-rate Q40 these will play about a tone delay unless you resample them with SOX.



compatible with samples for I2C, rather than make a cut-down version of SOUND specially for that system - not that I'm stopping anyone else...

Like all such projects, by the time I'd got QLSSS SOUND working I'd thought of a load of embellishments that could be added with relatively little effort. Therefore I've written not one but NINE new 'drives', SOUND1 to SOUND9, capable of replaying samples at various rates and with varying amounts of fidelity.

To use it, simply LRESPR SOUND4_BIN. Everything should work as normal except that you'll have nine new devices, SOUND1 (the default, as used by OPEN #3,SOUND) to SOUND9. The most useful of these are SOUND1 and SOUND2. If you COPY or PRINT eight bit unsigned samples to these devices, they will play in mono (same both sides) or stereo respectively, at the default SSS rate of 20 KHz. The other devices are cheap and cheerful attempts to play samples at rates other than the default. None of them are a proper substitute for involved sample-rate conversion and filtering - they introduce clearly audible distortion, in proportions which depend on the data - but the results might still be preferable to hearing the original sample at half or double speed, or waiting for laborious C-coded FIR resampling, FFT filtering and resynthesis with SOX (etc).

Macs, PCs and Unix boxes typically use sample rates of 11, 22 and 44 KHz. The middle one is close enough to our 20 KHz that samples play recognisably, though not quite in tune with other instruments. My code can also play at 10 KHz and 40 KHz, even though the QLSSS was only designed for the Q40's 20 KHz hardware rate. It can also dusplicate mono data through both channels.

Odd numbered SOUND devices are for mono output. Even numbered ones are for stereo replay. SOUND3 and SOUND4 attempt to play 10 or 11 KHz recordings by interpolating one extra value between each real one. This is just the average of the ones before and after, filling in the wave with straight line segments.

Some unwanted noise is thereby introduced, because you really need to take account of several samples either side of the point you want to guess at. The noise is at a relatively high frequency and hence obvious, especially as frequences from 5 to 10 KHz are not in the original data but may be in the interpolated output. The Amiga's hardware low-pass filter helps to mask these - POKE 12574721 with 2 to turn this on, and 0 to turn it off, or turn down the treble on your Hifi to filter the noise manually.

SOUND5 and SOUND6 play samples at half (rather than twice) their originally recorded rate. They do this by averaging two samples (for mono or stereo) truncating the result to stay in eight-bit range, and playing the result. Aliasing noise is likely, because frequency components above the SSS limits (from 10 to 20 KHz) are 'reflected' to lower frequencies, but it doesn't sound bad as long as the original recording was not particularily shrill.

The other lazy way to reduce the sample rate without reducing the pitch is to 'decimate' the data, skipping some items entirely. This is likely to sound worse than averaging unless the input is already filtered to eliminate frequencies above 10

```
1000 COPY thisplanet,sound3
1005 FOR i=1,1,5,7,3:COPY quack_ub,"sound"&i
1010 COPY weird,sound3
1020 COPY trumpet,sound3
1026 FOR i=0 TO 3:COPY groove,SOUND1
1028 COPY marvelous_ql,sound3
1030 FOR i=1 TO 5:COPY droplet_ub,"SOUND" & i
1040 COPY dwayne,sound3
1050 COPY wildeep_ub,SOUND1
1060 COPY wildeep_ub,SOUND1
1070 FOR i=1,1,3:COPY indigo_ub,"sound"&i
```

This little BASIC program plays a mixture of speech and sound effects at various rates. The default device (set with Toolkit 2) is assumed to be RAM. Many of them use SOUND3 as they were sampled at 10 KHz for I2C originally, but the 'Quack' sound is deliberately repeated and played at several rates in rapid sucession, to semi-musical effect - the SuperBASIC FOR syntax in line 1005 made this very easy. Groove is a loop (from the Black Velvet single) and plays seamlessly when repeated to SOUND3 with a simpler loop. As the name suggests, the result is a cacophony of quite interesting noises. The _ub sounds are from Mark J Swift's Mac-derived AlertSNDs.zip, downloladed from **http://pages.unisonfree.net/mswift/**, plus a few sampled from BBC SF programmes Red Dwarf and Doctor Who and converted from IFF-8SVX format to unsigned bytes with my BASIC routine. They are in the collections Simon gave to Ray and Dilwyn.

KHz. SOUND7 plays mono decimated samples at half speed by skipping alternate samples, and SOUND8 plays stereo decimated audio. You might compare SOUND6 and SOUND8 when playing CD audio - unless the original was from AM radio, old tape or vinyl, it will sound a lot better filtered and resampled with SOX, but at least it will be at (about) the right speed when played this way.

SOUND9 is the only really decent way to play 40 KHz samples, but it only works in mono. It plays alternate mono samples to left and right channels, so if your right speaker is slightly further away than the left - allow about half an inch, for the distance sound travels in air in one twenty-thousandth of a second - the two should reach your ear(s) neatly interleaved at 40,000 Hertz. You might need to adjust the speaker locations to suit your altitude and the weather, and clamp your head in a vice for perfect results, but even if you're sloppy about it, this really works.

SOUND channels support the FPOS toolkit function (assembler coders can call FS.POSRE with an offset of 0) which reports the amount of queued sound, as returned by the SSS_SAMPLE call. This might give results varying by a factor of two from expectations due to an ambiguity in the spec - is it samples or sample pairs that it counts? Jonathan Hudson's PLAYSOUND contains a fudge-factor to accomodate this bug in early SMS versions. If necessary, suck it and see.

In addition to the source and binary code, documentation and sample samples (sic) the SOUND device is supplied with DIRECT_SYNTHESIS_BAS - a program to generate and play simple sine waves. Adjust the SOUND device name in line 150 to hear the waves converted in various ways and rates. You can write a complete synthesiser this way - Yamaha DX-series FM synths use the same principle - although best results are typically obtained by mixing pre-recorded samples and computed waves. A lot of modern commercial synths work exactly this way.

Limitations

The SOUND device has a few known bugs and infelicities. I welcome suggestions on how to improve these. OPEN_IN is not rejected, though it makes no sense. At the moment all types of OPEN are treated the same way, allowing single and multiple byte output.

Output will glitch periodically if sound interrupts are missed. This is most likely during disk access, so play sound from

```
100 REMark QLSSS - Direct sound synthesis
110 REMark Tuning routine for Paula audio
130 REMark Demo by Simon Goodwin Jan 2001
150 t$="":s$="" : OPEN #3,SOUND1
160 FOR i=0 TO 255
170  t$=t$ & CHR$(127*SIN(i/128*PI)+128)
180 END FOR i
185 FOR i=1 TO 4:t$=t$ & t$
190 FOR i=0 TO 255 STEP 2
200  s$=s$ & CHR$(127*SIN(i/128*PI)+128)
210 END FOR i
220 FOR i=1 TO 4: s$=s$ & s$
290 per0=14676134:per1=per0+16
300 REMark Period @ 20 KHz (PAL) 177 (NTSC) 179
305 per=177
310 REMark Period @ 22 KHz (PAL) 161 (NTSC) 162
315 CLS :REMark Allow fine tuning by arrow keys
320 REPeat poll
330   k=KEYROW(1) : PAUSE 10
340   IF FPOS(#3)<10000 THEN PRINT #3,t$;s$;
350   IF k && 128 THEN per=per+(per<710)
360   IF k && 4 THEN per=per-(per>124)
370   POKE_W per0,per:POKE_W per1,per
380   AT 0,0 : PRINT "Period ";per,
390   PRINT 1E6/(.281937*per);" Hertz",FPOS(#3)
650   IF k && 64 THEN EXIT poll :REMark SPACE
680 END REPeat poll
970 CLOSE #3
```

This demonstrates the direct generation of simple waves, built in strings and output with PRINT. S$ and T$ are 78 Hertz and 156 Hertz mono sine waves. The loop at the end tweaks the 'period' registers in the Amiga's Paula chip to allow fine-tuning. Press up and down arrows to shift the pitch in small steps. The program shows the exact sample rate in Hertz (assuming a 15.6 KHz PAL display), and the amount of data in the buffer as it runs. Note the use of FPOS to check if the buffer needs re-filling, when it has less than 10,000 entries, and the PAUSE to prevent the loop wasting CPU time when no key is being pressed. Press SPACE when you've found the required rate, then use COPY or PRINT to output other samples.

RAM (disk, string arrays or via PEEK$) if you want long samples to sound good. This happens sometimes in Amiga Qdos Classic 3.25; it's most likely if the ACE screen updater is in use, rather than the Blitter. Mark J Swift has updated the VDU and ACE 'ROM' files in Qdos Classic to eliminate these glitches. Updates also make it possible to adjust the Amiga (only) sample rate, volume and balance (left/right), for easier listening and closer correlation to standard rates of 11/22/44.1 KHz.

If you stuff bytes into the queue which will not fit, the device should wait until there's room. Current low-level QLSSS implementations use a 200K buffer and start again at the start when it's been used, so a slight glitch or click is periodically likely if you play sounds that last more than five seconds. The SOUND device can't fix this - when QLSSS uses double-buffering the problem should go away.

The stereo, averaging and decimated devices expect samples in pairs (or sets of four). You'll get a BAD PARAMETER error if the string presented to the SOUND device does not contain a whole number of sets of samples. Also there's no provision (yet) to keep the previous values output from one string to the next, when averaging, so the converters revert to mid-scale at the start of a new squence. This will yield a one-sample dropout between chunks of sound, e.g. when COPYing. Expand the COPY buffer (as explained in some of my QL World articles) to minimise this effect, or use a different variant of the SOUND device.

I hope that someone (eh, Zeljko?!) will implement audio hardware with higher rates and resolutions for Qdos, but the existing systems are capable of good results. Even a cheap Amiga 500 supports all the SOUND drives, and has good quality stereo line outputs, so you don't need to splash out on a Q40 or Linux box; all the software is free, and source is available if you want to take it further. If you've got a standard QL your options are more limited, but it's still a lot better than BEEP and Minerva+I2C offers many benefits beside an intro to sampled sound.

Simon N Goodwin,
qdos@studio.co.uk

CAMBRIDGE Z88

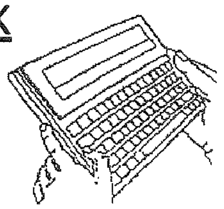
OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*
MOBILE: 07808 576118
E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) ✱



WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE, 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	$180
CAMBRIDGE Z88 COMPUTER	£99	$130
☆**RECONDITIONED Z88**	£60	$90
32k RAMPACK	£18	$25
128k RAMPACK	£38	$50
32k EPROM PACK	£18	$25
128k EPROM PACK	£24	$33
256k EPROM PACK	£55	$75
EPROM ERASER	£32	$43
PARALLEL PRINTER LEAD	£39	$50
SERIAL PRINTER LEAD	£12	$16
✱ **MAINS ADAPTOR (230vac; 6v,500ma)**	£10	
TOPPER (PROTECTIVE COVER)	£12	$16
CARRYING CASE (PLASTIC)	£10	$12

1/2 PRICE STARTER PACK 1 - Z88 ☆ PLUS TRANSFER KIT + 32K RAM + 32K EPROM £72 ONLY!!

FILE TRANSFER TO OTHER COMPUTERS

SPECIAL Z88-QL SERIAL LEAD	£10	$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	$3
PCLINK KIT (For PCs)	£25	$35
Z88 TO MAC KIT	£25	$35
Z88 TO BBC KIT	£25	$35

ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

Searching for a String (2)

John Sadler

As promised, here is the solution:

```
100 CLS: CLS#0
110 REMark INPUT "Input string to be searched ";A$
120 REMark B$ = Search String
130 A$ = "xyxxyxyxyyxyxyxyxyyxyxyxxy"
140 B$ =""
150 n = LEN(A$)
160 m = LEN(A$)
170 REMark Create an array to hold next comparison
180 DIM nxt(m)
190 REMark Get the initial 2 items
200 nxtstrg
210 j = 1: i = 1: n = 0  : REMark holds length of search string
220 PRINT " " & A$
230 REMark main loop
240 REPeat instrg
250  AT #0, 0,0: PRINT#0,"Input next character ";B$;"?"
260  in$=INKEY$(-1)
270  n = n + 1
280  IF CODE(in$)= 10 THEN EXIT instrg :REMark AscII 10 equals return
290  B$ = B$ & in$
300  IF n › 2  THEN getnxt(n) : REMark get next item in nxt()
310  findstrg           : REMark find next match
320  IF i › m  THEN EXIT instrg
330 END REPeat instrg
340 IF s THEN
350  PRINT "Index of match is "; s
360 ELSE PRINT "No match found."
370 REMark Procedure to find next match
380 DEFine PROCedure findstrg
390  s = 0
400  REPeat z
410   IF s ‹› 0 OR i ›m  THEN EXIT z
420   prtstrg i,j,A$,B$  : REMark print display to show algorithm
430   IF B$(j) = A$(i) THEN
440    i = i + 1
450    j = j + 1
460   ELSE
470    j = nxt(j)
480    IF j = 0 THEN
490     j = 1
500     i = i + 1
510    END IF
520   END IF
530   IF j = n + 1 THEN
540    s = i - n
550   END IF
560  END REPeat z
570 END DEFine findstrg
580 DEFine PROCedure nxtstrg
590  nxt(1) = 0
600  nxt(2) = 1
610 END DEFine nxtstrg
620 DEFine PROCedure getnxt(i)
630  LOCal l, z
640  l = nxt(i - 1)
650  REPeat z
660   IF l ‹ 1  THEN EXIT z
670 REMark prtstrg i,l,B$,B$
680 REMark prtnxt i,l
690   IF B$(i - 1) = B$(l) THEN EXIT z
```

```
700   l = nxt(l)
710  END REPeat z
720  nxt(i) = l+1
730 END DEFine getnxt
740 DEFine PROCedure prtstrg(i,j,x$,y$)
750  AT 0,1: PRINT x$
760  FOR x = 1 TO i
770   AT 1,x: PRINT " ";
780   AT 2,x: PRINT " ";
790  END FOR x
800  FOR x = 1 TO i-j
810   AT 3,x: PRINT " ";
820   AT 4,x: PRINT " ";
830  END FOR x
840  AT 1,i: PRINT x$(i)
850  AT 2,i: PRINT y$(j)
860  AT 3,i-j+1: PRINT y$
870  FOR x = 1 TO n
880   AT 4,i-j+x: PRINT nxt(x)
890  END FOR x
900 END DEFine
```

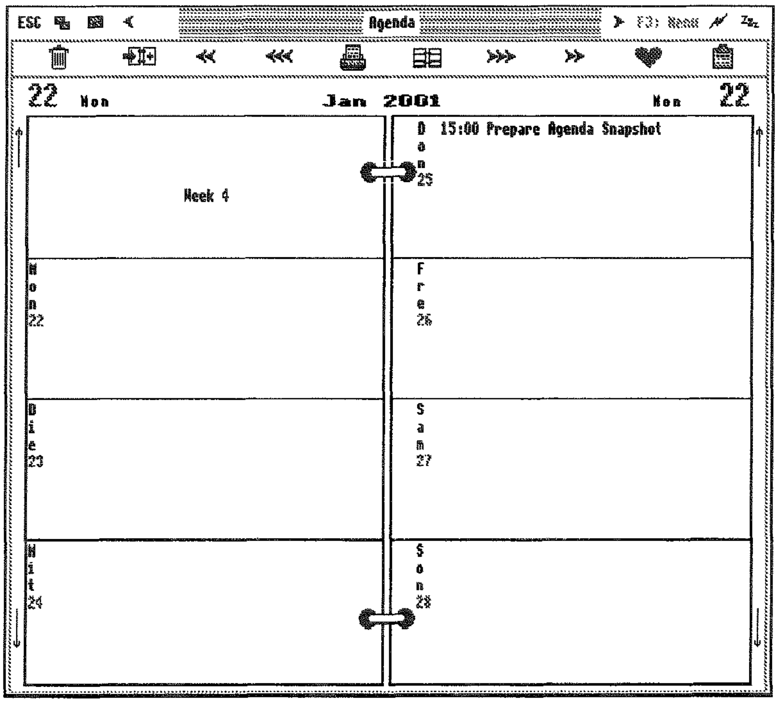
Agenda - A Review

By Timothy Swenson

Agenda is a program that I thought should have been written many years ago. There have been a number of productivity tools written for the QL, but a time scheduler has not been one of them.

Agenda is a computer version of a daily, weekly, and monthly planner. Paper equivilents are Day Timer, Geodex, and Franklin Planner. These are smaller little binders with pages for days, weeks, and months. Scheduled events are written down on the pages in the pre-printed time slots. When checking a particular day for events, a quick glance can tell you what is going on and when.

Agenda takes that paper idea and puts in on the QL, even showing little binder rings between the pages. The main display looks like two pages in a small binder, one side being the morning, and the other the afternoon. When events are schedule for a day, they appear on the page, in time sequence. Besides scheduling multiple events for each day, a To-Do list can be created for each day. If any thing on the To-Do list is not completed (deleted) today, it will appear on tomorrows To-Do list. Agenda also keeps a list of annual events, like Birthdays.



The Manual

As with all JMS software packages, an A5 manual comes with the software. Given that the concept of Agenda is fairly simple, and the program is not very complex, the manual is not very big. It covers all that is needed; installation, configuration, using, and backing up.

Installation

Agenda comes in two different "flavors", the standard Pointer Environment (also called WMAN), and Prowess. It also comes in German and English version. Given that the author is German, I can see why the two languages. Not being a Prowess user, I only tried out the WMAN version.

Agenda requires a few toolkits to be loaded at boot before it can run. All toolkits that are required come on the disk, except for ToolKit II (TKII). The toolkits are:

- Qmenu
- QLib runtimes
- DataDesign database engine
- DataDesign SBasic interface

The Prowess version only requires the standard Prowess setup (which includes DataDesign), but a new Prowess SBasic Interface binary is needed (and comes on disk).

Since I already had Qmenu and Qlib loaded, it did not take me long to figure out what binaries I needed off the disk, to be put into my BOOT program. I then copied the Agenda executable to my hard drive. I had to rename it from the long "English_WMAN_Agenda_EXE" to just plain "Agenda_exe".

Configuration

Agenda is configurable via MenuConfig and allows a number of different options to be configured. The three main items that you must configure are: location of data files, time format, and date format.

You are allowed to put Agenda and its data files anywhere you want, but you need to tell Agenda where to find it's datafiles. For me, I just created a 'win1_agenda_' directory and put everything there.

The hardest part of Agenda is remembering what format to enter dates and times. There are four different time formats and three different date formats. There is no reminder of either format when entering data for a new appointment. I found that a little yellow sticky on the edge of the monitor was a nice way to remind myself. The different data and time formats are designed to allow customization for different countries.

Other items that can be configured include choosing color (WMAN), choosing font sytle (Prowess), and choosing the date separator.

Using Agenda

Using Agenda is simple as the concept is fairly simple. Enter appointments or meetings and view your schedule. There are clickable items that allow you to move forward or backward a day, week, or month. You can view a single month (without appointments listed) to see the relation of the days (Is the 5th a Tuesday?).

When Agenda is executed, if there are any annual events (Agenda calls them "Birthdays & Special Days"), or any To-Do items, they are displayed in a popup window. You have to click on OK to get to the main part of Agenda. Agenda also pops up with the "pages" on the current date.

One feature that I would like to see added to Agenda is an Alarm. Basically, you could tell Agenda to alarm you (beep) a 1, 2, 5 or so minutes before an appointment. As is, Agenda reminds you about daily events (when it is first executed), but a reminder for hourly events would be nice. Maybe this could be added in the next version.

There is a nice discussion in the manual about the two different ways to display meetings. The first idea is to have a set space for each hour of the day. So, if you had an hour meeting at 9:00 and another hour meeting at 11:00, you would see a gap display in between the meetings. This looks nice and is easy to visualize the day like this, but doing this take up a lot of screen space. Users with the standard QL screen would be spending a lot of time scrolling up and down to view the day.

The author decided to display each meeting one after another, so the 9:00 meeting and 11:00 meeting are shown next to each other. This saves on screen space, but does deviate from the "standard" way the hardcopy day planners do it.

Additionally, no matter how long the meeting is, it is displayed on Agenda as one size. Since this does now allow the user to quickly see how their time is scheduled. A simple solution would be to list both the start and finish times on the display. A quick workaround is to constantly write down the end time yourself in the area where the meeting is described.

The program integrates well with the PE desktop. It has all of the typical PE options (move, resize, sleep/button). I can see Agenda being part of a boot program and running as long as the QL is turned on.

Conclusion

Given that the day planner concept is fairly simple, Agenda is a fairly simple program to use. Getting it to be this simple and look as good as it does, is not a simple matter. The program looks as a professional QL program should. Making the main display area look like a day planner by having two binder rings is a nice touch.

It could stand a little tweaking to improve it's display to the user, but over all the program does a good job. For those that use the QL almost exclusively and have different meetings or appointments to schedule, this is a must-have program.

RWAP SOFTWARE

Flashback SE v2.03(Upgrade) £5

This is the ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately, only available as an upgrade to the original version (Original still available from Sector Software).

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT return, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of current tax year rules.

Q-Help v1.05	NEW VERSION	£10
Q-Index v1.01	NEW!!	£5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.
The PD toolkits referred to are available for £2.
Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and fnd out the commands which relate.

Sidewriter v1.08 £10

Produces landscape printouts of Easel/QSpread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix up to inkjet printers. A most useful utility written by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.01 £8

New improved printer drivers, providing up to 720 dpi printout in full colour from all programs written for use with ProWesS, such as LineDesign and Paragraph. Work on all Epson inkjet printers which support binary mode compression (740,850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26	£20
Genealogy For Windows	£50

Keep track of your family tree! Add individuals, with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family for future generations. QL version now supports FileInfo II and QMenu as well as keeping details of both the male and female trees. PC version is event driven - enter the details as they appear in documents and it generates the tree from these. Both programs easy to use with step by step tutorial. QL data and GEDCOM can be transferred to the PC version.

SBASIC SuperBASIC Reference Manual £40

Updates £6 each, £10 for 2 (Current Version - Rel 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on a standard QL, plus the keywords under SMSQ/E, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date by regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note Price for the book does not include postage and packing).

Return To Eden v3.08	£10
Nemesis MKII v2.03	£8
The Prawn v2.01	£8
Horrorday v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

Classic QL adventures, now re-released without any need for microdrives. These include mainly text adventures, catering for all tastes, from the spoof Prawn, through to a Hammer Horror, fighting the bad-guys in the old West and battling with robotic hoards and goblins. Return to Eden is a massive three disks of adventure, with pictures for each location and a captured prince to rescue. With three characters to control, each with their own abilities and skills, this one should keep you amused for many an evening.
All six adventures are available together for only £25.

Image D v1.03 NEW!! £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

Q-Route v1.08C £25

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

QL Cosmos v2.03 NEW VERSION £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

A range of games to keep you amused on the QL. Some are old favourites, like Golf and a quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for?
All 6 programs £28 only.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5
FlightDeck v1.04	£10

These are the latest maps for Q-Route (now at v1.08C). Find your way around the various countries covered. South and West Yorkshire Map is a much more detailed area of that beautiful part of the British isles.

Britain map v1.10	£2
BIG Britain Map (needs 2MB) v2.01	£5
South & West Yorkshire Map v1.04	£1
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.02	£2

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8
War In the East MKII v1.24 (Upgrade Only)	£10

For the wargaming enthusiast - D-Day is a classic table top wargame, where you control either the Allies or the Axis forces and play against either the computer or another human player. With the ability to define your own army set ups and a choice of four different scenarios, this should keep you entertained for a while. Grey Wolf places you in charge of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

RWAP Software, 4 Anvil Crescent,
Coseley, West Midlands
WV14 8GA
TEL: 01902 836888

Cheques in £sterling payable to 'R.Mellor'

* Also known as Trading Accounts

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS

Well, as predicted in the last column, my wife has finally given birth to our new daughter, called Barbara, and things are gradually beginning to return to normal here at QBranch Towers - apart from the sleepless nights that is. Thanks to all who offered congratulations and sent cards.

Open The Pod Doors Hal

So I imagine that you are now all heartily sick of both turkey and Arthur C. Clarke / Stanley Kubric quotes but, for us, this does have a degree of relevance. Science fiction writers and pundits have long used the 2001 milestone as a marker for an era when science will have reached a degree of sophistication whereby the general public will be jetting off for their summer hols on Venus and we will be having conversations with our computers. How ironic, therefore, the papers, magazines and news programs of the later part of 2000 should be so full of talking heads predicting the end of the 'e-society' and the only conversation the bulk of the population is having with their computers is unprintable and uncomplimentary to computer vendors.

I will stray from my usual realm of QL matters to talk a bit about the state of computational science in the dawn of the 21st Century this being the 'pedants millennium'. I say this because of the continuous arguments about there being no year zero forgetting that the date which started the whole thing was probably dreamt up by a Roman Emperor more interested in the cohesion of an empire than any biblical reality. For computer users we might as well call it 19 AD (after DOS).

The Sublime Disillusionment of Experience

Through the last years of the Twentieth Century we were bombarded with adverts and exortations to buy the latest computing equipment and these ads did not go just to the computing fraternity they were broadcast willy nilly to the public who gulped it down in great lungfulls. People who had difficulty in reading tabloid newspapers were told that they had to buy a computer and have the most powerful available otherwise they would be struck off from society and consigned to the dark pages of prehistory. Of course, having got a computer, most did not know what to do with it apart from downloading online pornography and, in the end, it was like selling a high performance car to someone who does not even know that you have to lift the bonnet from time to time to see if there is any oil in the engine.

Computers were marketed like fridges or TVs and, if you had bought a fridge, and then come downstairs one morning to find a big wet patch where the fridge had stopped working because you have put the peas in the wrong compartment or bought the wrong ice cream or loaded it with an out of date pack of frozen waffles you would be as upset as these people were when their shiny new system stopped working. Luckily there is not a 'Fridge User' magazine with packs of food on its cover put in your new fridge or the cooler would soon be stuffed with as many useless things as the average PC.

The Faster We Go....

At some point in the vast Hurdy Gurdy of the e-hype it dawned on the more cognisant of users that no matter how fast the processor was the applications still crawled along. Some of the problems were hardware based because the CPU got faster but the RAM stayed at the same speed. A bit of a boost occurred when the newer ATA 100 hard drives appeared but the problem lies more with the fact that applications themselves just continue to get bigger and more bloated with 'features'.

All very well you may say but try asking the average PC user about the features of, say, a word processor. You will find that the bulk of the features mentioned can be found in Perfection or Text 87 and they rarely stray further from these. This, course, brings me to the next point.

Where did all the manuals go?

No one supplies paper anymore. Maybe we all want to save the forest but that is not the real cause of the lack of instructions in software. It is far cheaper to not have to supply a manual - it is also far more profitable to convince Joe Public that he does not need a manual (after all he does not read them does he?) and what is even better is that he is so busy being cool with the 'online talking paper clip of destiny' that he has not noticed that the software house has not reduced the price. So you get 650 Mbs worth of code and online manual and you use 5Mbs of it. Not only that but you paid a fortune for it - unless of course you did what most users do and copied your friend's CD. They rip you off you rip them off what could be more natural ?

The Paperless Orifice

This whole fixation with the online, onscreen, on message public should have led to the paperless office years ago but why is it, therefore, that we are still drowning in pulped trees? One advert for a printer that I saw in a recent magazine said 'What is the first thing you do with an email when you receive it? - Print it of course' Why? What is the point of that? It is on the screen, on the hard disk, you have read it. If it is important THEN print it but not all of them. This is not just advertising hype. I know people that do this.

The other end of the scale it seems that people are abandoning the internet rooms of colleges and universities in favour of the libraries. Yes, those places that would all be closed down by now according to the e-pundits because all of the books and papers will be online. Eyestrain caused by staring at monitors is one reason for this but you cannot curl up in a chair with a monitor so you either sit bolt upright in from of a VDU or - you pick up a book from the library.

Now I am not some throwback luddite cursing technology but I do believe that there is a place for both paper and code in the world and that neither should necessarily displace the other. When Rich Mellor and I talked about the SuperBasic Reference Manual at the Hove Show some people said we should not release it on paper but electronically. I am sure that those of you who have the tome get more out of it than you would have if we had done that.

So Why are you still Here?

Small wonder then that we still have a hard core of users even if many of these own both a QL and a PC. Purists amongst us would say that we should not have a PC but this is the same old boring attitude of red team versus blue team that I talked about last time. This is the attitude which launched a virus reported on the internet recently (or it could be a hoax - on the internet no-one can hear you scream). This little virus does the usual trick of emailing itself to all of the names in your address book and then changes the names of files on the hard disk. It leaves a text message saying Isn't it about time you upgraded to LINUX?' and is signed by 'The Penguin'. dumb or what?

I have been through the reasons why many of us still use the QL and its native hardware before so I won't plough that furrow again but there are those amongst us who have proposed that we re-promote the QL as a platform. Many suggestions have been made as to what direction this should take and these range from a children's game machine to a PDA (Portable Digital Assistant - sounds like something you can buy in one of those shops with black windows).

I was recently contacted by Gordon Laing who writes the 'Retro' column in Personal Computing World. He wrote a very nice piece in January's edition of the magazine about the QL which was, on the whole, very positive. He talked about the origins of the QL and mentioned all of the major developments including the Q40. He even gave out my website address and I got a few nice emails but they were mostly nostalgia and a couple of offers from people who wanted to sell or give me an old QL. Only two people, however, wanted to buy back into the QL one bought one of my stock of second hand QLs and the other upgraded from QPC 1 to the new QPC 2.

What does this mean for ideas of boosting the QL? I must confess that I do not know. Do the readers of PCW only buy it for the adverts and leave all other stuff unread? I suspect that any system without a terrabyte of RAM and a hard disk the size of a small Africa state is regarded as kids stuff and I am not too sure how to change this. I am not even sure that it can be changed. Peoples attitudes are now so entrenched that they have no real way to see anything else.

More Hz Less Speed

Working, as I do, in a place which sells the latest PC hardware we do get a chance to test all of the latest stuff when it arrives on the market. We recently tested a 600MHz Duron processor (cost £45.00) against a 1.1GHz Athlon processor (cost £350.00) the speed gain running a benchmark called 'Mad Onion' was about 30%. We told one of our customers this when he ordered a new system form us - guess what? He bought the Athlon. I am no great fan of benchmarks because I think that they just prove how fast the benchmark programs run but I do think that this shows up they way people think. Go for the high numbers boys!

Postcards from the Hedge

When the QL 2000 show was being arranged the traders all got together and sent out the usual flyers to inform people about the show. We do this on a regular basis and have a large database which is maintained by Tony Firshman. We decided that the time had come to try to refine the database and try to weed out those who do no want to have flyers sent to them. We included in these flyers a postcard which asked the recipients to fill in and return if they want to stay on the database. We even offered a prize to the person whose postcard was picked at random at the start of the year. The response was very disappointing. We had fewer returns than we have readers of the magazine and that cannot be correct. A lot of people will have thought 'I can't be bothered with all of that' or may have intended to send it and forgot but we do need to get the information in. This database is not just for advertising purposes. It is used to get information to people who do not subscribe to the magazines and to keep them informed of developments in the QL world. If you have not filled in the postcard or know QL-users who are not on the database do your best to get them to pass their details to Tony Firshman either by post at:

T.F. Services
29 Longfield Road
Tring
Herts HP23 4DG

or by email at
tony@firshman.demon.co.uk

We do need to get this information organised and it is in your interest so get on the list and get informed. Please supply email addresses where available.

How many LINUX programmers does it take to change a light bulb?

None - They decide to form two groups. One group tries to port a bulb in from another platform while the other tries to write a patch for the broken one. When they have finished they apply to Linus Torvald to have it approved but find it is already daylight.

How many Windows Users Does it Take to Change a Light Bulb?

None - they have been in the dark this long why bother?

Computers at the Dawn of Time

Now we are here finally in the new millennium I wonder what would have happened if computers had been around earlier in history. Here are a couple of thoughts maybe you can supply more.

The Three wise men, for instance, could have used their GPS systems to get to the stable faster but then Joseph would have gone on the internet, tried to book a room, seen it was full and stayed at home. And they would have ordered their gift via the online shopping so they would either not have arrived or have received the wrong items.

And Good old Albert Einstein. Would he have produced his theory of relativity quicker if he had access to a PIII system or would he still be trying to re-install Office to write it up.

'God does not play with mice.' as he probably would have said.

At Last a bit of QL Stuff!

An email last week to the QBranch HQ had me puzzled. A User who had bought QPC2 from me said that the computer locked up whenever he ran his BOOT file. I suspected a problem with the BOOT file itself and I suggested that he tried rewriting the BOOT file. 'I Cannot do that,' he replied 'because the BOOT file is on the hard drive and the computer locks up as soon as it starts.'

I suggested that he inserted a floppy disk into the A: drive with a boot file which just contained the line:

```
100 STOP
```

He had not thought of that and it gained him access to the computer so he could look at the BOOT file. The problem was one I had not expected. He already had an Aurora and he used the same BOOT file on QPC that he used on his Aurora - including the line which loads SMSQ/E. I have had other people who have upgraded to the new versions of SMSQ/E for the QXL and QPC asking if they can use these files on their other machines. The answer is a resounding 'NO'. Each version of SMSQ/E is designed to run in a specific environment and will cause problems if you attempt to use it in a different way. In this case it was a complete lock up and, since SMSQ/E is LRESPR'd at the start of the BOOT the effect is instantaneous.

It is a tribute to the ingenuity of Tony Tebby that we are able to use the same commands on so many different platforms without a whole load of conflicts.

All the Print That's Fit For News

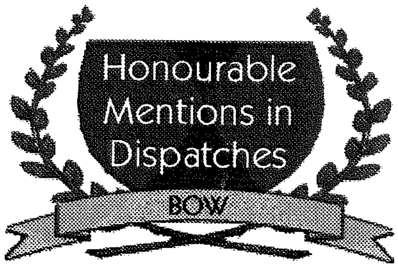
I have also pondered in the past about the kind of questions I would get when the colour drivers for the other platforms became more widespread. I was not prepared for the question I got last week. A User emailed me to ask why QPC2 does not print to his LEXmark colour printer now that it ran in a Windoze window and had 16 bit colour. Now you all know why, don't you, because you have been reading this magazine and it has been explained to you but I did not expect someone to assume that because the screen has colours and he has a colour printer that the two would go together.

There was talk on the user group of writing a Postscript driver which would talk to the Windoze printer under QPC2 but I must confess that I am a bit puzzled here. Why chose Postscript. I realise that is is a very concise way to produce graphical output (although I have never tried it) but surely the way to do this is to write a QPC2 driver which talks to the Windows Print Manager and allow that driver to accept input from a small range of standard printer drivers.

Epson ESC/P2 and FX-80 (for the older programs) would be a favourite one to choose since almost every QL program can produce output in this format. If we invented a print device (say PRI1_) as the portal for this new device and it installed itself when LRESPR'd in the boot file we could tell all of the programs to print to that using an EPSON driver and then it would pass output to the Windows Print Manager and that to the printer. Probably a bit slow but a solution nonetheless - and one which does not need a Postscript driver to intervene. It seems a classic case of reverse engineering. If you know the codes that EPSON uses to produce coloured output from a program then you must be able to interpret those codes to produce the kind of input needed by the Windoze Print Manager.

OK all you experts tell me why it does not make sense.

And, that leads me to this years first....



Honourable Mentions in Dispatches

Yes you've guessed it. This issues Honourable Mentions have to go to Marcel Kilgus for his excellent work with QPC 2. I won't dwell too much on this because I am sure that there will be as much of a buzz about it in the rest of this issue as there was on the user group mail list where it was first announced but it really is a stunning example of why the QL scene is often at the opposite end of the spectrum to the PC world. When a new version of a piece of PC software emerges it is often twice the size, half as fast and with few noticeable improvements. QPC 2 v2 however is only slightly larger than version 1, It is much faster on all the platforms I tried it on, and packed full of improvements.

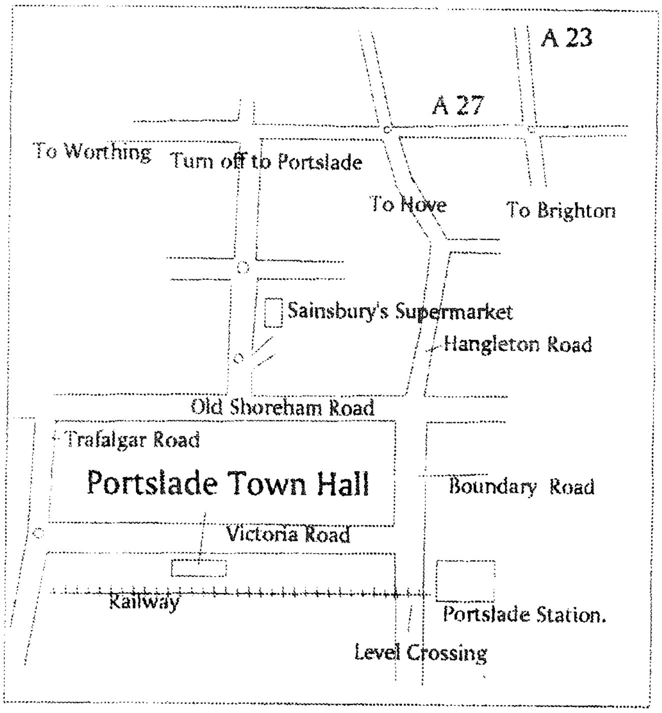
Not just the colours, which are good enough, but new DOS device which enables QPC 2 to share files with the PC completely transparently. Of course we run into the old file name character limit (down boys!) but it is fairly simple to have a directory on the root device in the DOS side of the hard disk to use as a temporary storage area for files waiting to be transferred to QPC 2 or which have been transferred from QPC 2 and are awaiting filing. I do this on the PC anyway because I have both download and upload directories for files to and from the internet.

Great Stuff Marcel - wear your medal with pride !

You've read another very interesting QL Today (we hope). Articles from Users for Users... wouldn't you like to see YOUR article being printed? Let other users share you experience! You do not need to be an expert, every level of knowledge is fine. Or write a letter, tell us about your view, your system, your plans for the QL Future!

Hove Show

6th Annual Sussex User Group/
Quanta Workshop
Portslade Town Hall
Victoria Road, Portslade
Sunday 4th March
10 am - 4 pm



This is the sixth year we have run this show and we are trying a change of venue from the usual Hotel location to my local Town Hall. This is a much larger venue with better parking and easier access. I have also arranged a bevy of local ladies to do some catering so the food and drink situation should also be better.

We expect all of the usual QL culprits to be there. T.F Services, Jochen Merz, W.N. Richardson, Just Words, and QCelt Software. Our user group will have a bring and buy stall and Enrico Tedishi will bring his display of Sinclair Memorabilia. Our User Group has recently been given a complete set of QL World Magazines and Quantas and these will be available for free so you can fill in any gaps in your collection.

Hotel details will be available in the flyer which will be sent out in February. If you have not returned the postcard from the QL 2000 flyer please contact Tony Firshman at T.F Services to make sure you are on the mailing list.

See you all there.

QL Show North America

Quanta and NESQLUG are proud to sponsor the North American QL2001 Show. It will be held near Montreal, Canada on Saturday 2 June 2001. The local host is Francois Lanciault who developed Paragraph, the WYSIWYG Word Processor for the QL. Several QL vendors including QBranch, TF Services, J-M-S, and Q-Celt are making plans to come.

There will be a short NESQLUG meeting with all invited: Rigel Cable and Al Boehm will explain NESQLUG-NET, a linkage of member's web pages on a variety of QL topics. Dorothy Boehm is the focal point for tourist activities for wives and children. So bring your family.

Montreal is only 1 hour 10 minutes by car from the US border. For Americans crossing the border by car is easy - just have some form of ID with you. There is no trouble driving even though some of the signs are in French and speed limits are in kilometers per hour. Public transportation - planes, trains, etc - going across the border will ask for a passport.

Montreal has two international airports: Dorval airport just south of the city and Mirabel Airport just to the west of the city. Either is close to the show site. The closest big airport in the US is Burlington, VT., 40 miles south of the border. Plattsburgh, NY, is even closer to the border - 23 miles, but is a smaller airport with less flights.

Information on the recommended motel will be made available shortly.

Francois Lanciault has been stressed by the illness of his son, who at last report is getting better. So until 1 March, please direct any questions to Al Boehm, tel: 256 859-8051, email: **albertboehm@juno.com**